Table Of Contents

Foreword

Congratulations on selecting the AEA AVT System... The Ultimate Amiga Video Terminal.

The AVT System is another quality product developed to address the needs of the video oriented Amiga user, and the amateur radio operator. It should provide you with years of operating enjoyment whether you're just starting out with SSTV and FAX or have been working with video since the modes were conceived.

The AVT System can function as a superb stand alone video transceiver over the telephone lines. Fast and dependable full color images transfer over the telephone providing functionality that typical "office FAX" systems cannot match. This, combined with the system's capability to fully automate these transfers if you have ARexx installed, provides the ability to utilize the lowest telephone rate windows running completely unattended at both ends of the telephone line.

To fully enjoy the benefits of the AVT System, please remember to read this user's manual carefully either before operating the unit, or relatively soon after you have it installed and have experimented with it somewhat. There are many, many features that are not "obvious", and without completely reading this manual you will be unlikely to run across them. If you have any questions once you have read the manual, we encourage you to contact one of our technical support representatives at:

Advanced Electronic Applications, Inc.
P.O. Box C2160
Lynnwood, WA 98036-0918

Voice - (206) 775-7373
FAX - (206) 775-2340
TELEX - (206) 697-2496

Please complete and return the warranty registration card packed with your AVT System. From time to time, we will make updates available to you and it is *important* that we have your correct mailing address.

TradeMark Information

Electronic Arts:
Deluxe Paint, Deluxe Paint II, Deluxe Paint III, Deluxe Photo Lab

MicroIllusions:
Photon Paint, Photon Paint 2.0

Commodore Business Machines:
Amiga, a500, a1000, a2000, b2000, WorkBench, Intuition, Exec

William Hawes:
ARexx

Rick Stiles:
UEDIT

Black Belt Systems:
CoComm, BoardMaster, Electron, Cross-05, LaserPlot

AEA:
AVT System, Amiga Video Transceiver, QRM mode, PK-232

Logical Devices:
PM68

Haitex:
Haitex 3-D glasses

Compuserve Information Network:
Compuserve, AmigaForum, AmigaTech, AmigaVendor, AmiagArts

Robot Research:
1200c, 450c, 400c, 400.

Microcraft:
Videoscan 1000

Motorola:
MC68000, MC68010, MC68020, MC68030

InTel:
i8080, i8086, i80286, i80386

Credits

We would like to take this opportunity to thank the following individuals:

Scott Gray, N7CTF (extra)
For his efforts in Alpha and Beta testing the AVT system, and carefully proofreading the documentation.

Deb Davis, N7IHY (general)
For her efforts in *very* carefully proofreading the documentation.

Peter Gerardi, KE4TP, ex WB4YQK (advanced)
David Saunders, WA4VWU (advanced)
Tim Heffield, N4IFP (advanced)
Gary Johnson, WA4RBI (general)
Johann Gyllenband, N5CST (general) ✓1-903-465-1454
For their efforts in Alpha and Beta testing the AVT system.

Dr. Anne Williams, N7LWZ (general)
For her unflagging support of the AVT project and its designer.

AEA, specifically **Mike Lamb, N7ML** (extra) and **George Buxton, N7EZJ** (technician), for having the vision and the courage to step away from the world of the compatibles and the clones and into one of power, elegance, and sophistication - the *Amiga*!

The AVT system was conceived and implemented by Ben Williams, AA7AS (extra).

Introduction

The AVT System is a combined hardware/software product that enables the Amiga user to send and receive images over either the telephone or a radio (Ham Radio, etc).

Images are not sent using "digital" methods, like your computer's modem or a packet terminal unit does. Instead, the images are converted to a digitally synthesized *analog* format which takes far greater advantage of the data bandwidth on any standard voice width channel, such as those mentioned in the previous paragraph.

The system can send a full screen color HAM (Hold-And-Modify) image in as little as 12 seconds using analog encoding methods. These methods transfer the entire image, but provide a much more "general" view of it; the image is a soft focus type which is good for portraits, but not highly detailed subjects.

For more detailed image transfers, modes are available that take as long as three minutes for a full color, 320 by 400 transfer.

The system also supports analog facsimile. This is a very high resolution (1024 horizontal by 1200 vertical) grey scale image, as opposed to the color image modes detailed above.

Many of these SSTV modes are well known to radio amateurs throughout the world. The FAX modes that the AVT System supports are the same ones that the wire services, such as United Press International (UPI), use to send images around the world using shortwave radio. And, yes, using this system, you can receive these images directly, as they are being sent by the correspondent in the field... the news as it happens! The AVT System supports nine different methods of receiving FAX imagery.

The color modes are known to Hams as Slow Scan Television, or SSTV for short. The AVT System supports more than forty(!) different modes, and more are added as we learn of them. Currently, all known commercial SSTV modes the world over are supported by the system, as well as most of the experimental modes.

The AVT System hardware consists of a compact unit that is plugged into your Amiga at its parallel port, or a parallel switch connected to that port. There are no controls on the box: all hardware control is done via the AVT software - and there is a **lot** of controlling to be done. The AVT hardware has an RJ-11 jack (similar to a modem or modular telephone jack) for connection to the telephone system, two audio output connectors, a power connector (AC adaptor supplied), five audio input connectors, and a parallel connector that is for a cable to your Amiga's parallel port.

For telephone use, when the Amiga in question is running ARexx (a third party software package for the Amiga; see glossary), the AVT system is smart enough to detect the phone ringing, pick up the phone, switch to any video mode automatically, and transfer the incoming image into memory (or disk), then hang up; all before one minute has elapsed. On the other hand, it can also utilize ARexx to pick up the phone at a pre-programmed time (like when phone rates are low), touchtone dial a target AVT system, and send an image. To put it simply, this system is "the whole ball of wax."

Advanced features? Certainly! Specialized image processing modes that are specifically designed to handle the types of damage that an image may sustain during transfer over the airwaves or a poor telephone connection, error correction and lost data synthesis are just the tip of the iceberg when it comes to extensive image handling capability.

Full Amiga text font support, including re-assign and re-load font libraries, no limit on the number of fonts (other than available memory), italics, bold, underline, mixed, colored, color "sweeps", text undo, area fill, text preview, etc. are provided.

The AVT System also boasts a superb ARexx interface, immediately coupling the system's capabilities with any other Amiga application that supports ARexx (like databases!). ARexx is an optional language package you can obtain for your Amiga system from your Amiga dealer.

ARexx has several specific enhancements it provides to your Amiga when you install it. First, it allows you to write stand alone programs, much like the BASIC language does. Next, it allows any individual program that supports it (as the AVT does) to be able to run macro commands, or even longer scripts of commands. Lastly, when you have more than one program running in your Amiga that supports ARexx, these programs can communicate with each other - this can be extremely handy!

Just one example is when you have a database program running as well as the AVT. It would be very easy to set up a script to ask you for a callsign, and then provide you with a list of all the pictures that person ever sent you - even load them into the AVT automatically. For the middle to advanced user, ARexx is a must!

The AVT runs from either the CLI with an abundance of "command line switches," or the WorkBench, using the "tooltypes" in its icon for a complete custom environment you set up, including full path specification and everything else you need to make it **your** program. It's *not* copy-protected and it allows you to navigate the Amiga's file system using one of the most sophisticated file requesters in any program today. AmigaDOS 1.3 and FFS compatible, the AVT System was designed especially for the Amiga - and it shows.

RF Interference (RFI)

The AVT Master has been certified under Subpart J of Part 15 of the FCC requirements. Operation is subject to the following two conditions:

1) This device may not cause harmful interference.

2) This device must accept any interference received, including interference that may cause undesired operation.

The unit generates and uses radio frequency energy. If the AVT Master is not installed and used in accordance with this manual, it may cause interference to radio and/or television reception. It has been type tested and complies with the limits of a Class B computing device in accordance with specifications in Subpart J of Part 15 of the FCC requirements which provides reasonable protection against such interference in residential installations. If the AVT Master causes such interference to radio or television reception (test by turning the unit on and off and noting any change in reception), you are encouraged to correct the interference using one or more of the following measures:

* Re-orient the receiving antenna

* Relocate the computer with respect to the AVT Master

* Plug computer into different AC outlet than the AVT Master

If necessary, you should consult with the dealer or an experienced radio/TV technician for additional information. You may also find the FCC booklet, "How to Identify and Resolve Radio-TV Interference Problems" a helpful resource.

The AVT Master *must* be used with peripherals that also meet the FCC's class B RFI limitations. Peripherals that are attached to your Amiga that are *not* FCC class B certified may cause interference when the AVT System is operating due to certain operations that the computer is performing internally.

Manufacturers are required to obtain this certification for any device that generates or uses RF energy, however many do not. It is *your* responsibility to ensure that peripherals that you attach to your Amiga have this certification.

Limited Warranty

Advanced Electronic Applications, Inc. warrants to the original purchaser that this product shall be free from defects in material or workmanship for ninety days from the date of original purchase. In order to obtain warranty service:

1) Complete and mail the warranty registration card within ten days to Advanced Electronic Applications, Inc.;

2) Send written notification to the address below or telephone as soon as possible after discovering a possible defect:

   Advanced Electronic Applications, Inc.
   Attention: Technical Support Department
   2006-196th Street S.W.
   Lynnwood, WA 98036-0918
   (206) 775-7373

The written notification must include a copy of the invoice, a description of the defective part or condition with details of the electrical connections to associated equipment, and a list of such equipment. Please enclose a separate sheet with your name, address, phone number, unit serial number, and date and place of purchase. Shipping charges for any parts or units submitted for replacement under this warranty must be paid by the purchaser.

Correct maintenance, repair, and use are important to ensure proper performance from this product. Carefully read the instruction manual. This warranty does not apply to any defect AEA determines is caused by:

1) Improper maintenance or repair, including the installation of any parts or accessories that do not conform to the quality or specifications of the original parts;

2) Misuse, abuse, neglect, or improper installation;

3) Accidental or intentional damage by the user.

All implied warranties, if any, terminate ninety days from the date of original purchase. AEA is not responsible for damage to other equipment or property, or any other consequential or incidental damage of any kind whether it is based on contract, negligence, or strict liability. Maximum liability shall not, in any case, exceed the purchase price of the unit.

The foregoing constitutes AEAs entire obligation with respect to this product. The original purchaser and any user or owner shall have no other remedy and no claim for incidental or consequential damages. Some states do not allow limitation of how long an implied warranty lasts or do not allow the exclusion of incidental or consequential damages, therefor, the above limitations and exclusions may not apply to you.

This warranty gives you specific legal rights. You may also have other rights which vary from state to state.

Hardware Installation

To install the AVT System, you need to make several cable connections. First, ensure that your Amiga systems power is off. This is *very* important! **Failure to turn the Amigas power off during installation may result in damage to *both* the Amiga and the AVT System, and voids your warranty.**

With the computer's power **off**, and no other connections made to the AVT System, plug the parallel cable that is supplied into the AVT hardware and into the Amiga's parallel port, or into a distribution connector on a parallel switch that is attached to the Amiga's parallel port. Ensure that the connectors are seated properly at both ends of the cable; they should fit squarely against the chassis where the connector is mounted. Any "tilting" or other visible misalignment indicates improper installation, and you must remove the offending connector and try again.

Once the parallel cable has been attached, connect the audio cables. You will just need one cable if you are only planning to receive, for instance if you are a shortwave listener (SWL).

The AVT hardware has five audio inputs, and you may use any of these. However, for the initial installation we suggest you use the input marked **Input 1** so that the steps in the tutorial will be exactly correct. The AVT System selects its inputs through your interaction with the software, and the steps in the tutorial assume you are using input number one.

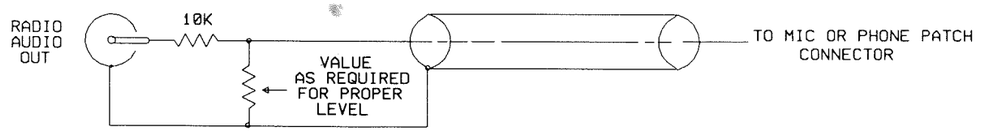
The inputs on the AVT system are all standard female "RCA" jacks. The outer shell is ground, and the center wire in the cable carries the audio signal. To connect this to a radio, you'll need to ensure that the signal from the radio provides at least 250 millivolts, with ground and signal available as just explained.

If your radio does not have a phone patch or tape monitor output, you can connect that cable to your speaker in most cases - however, this is not the optimum situation, because the setting of the radio's volume control will affect the quality of the received video.

If you will be transmitting video, you'll need to make at least one more cable connection, and you may make up to two more depending on how you will be using the system.

The first connection, which is required in all cases where you will wish to transmit using the telephone *or* a radio, will be via a male-to-male RCA cable. Connect this from the Amiga's **left audio output** jack at the rear of the machine to the jack on the AVT hardware that is marked **AMIGA AUDIO.**

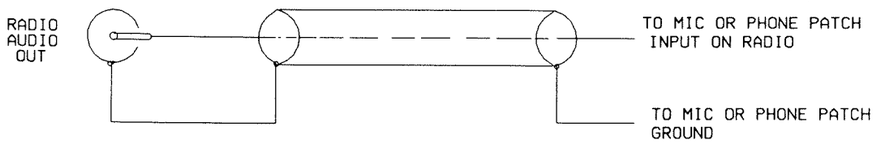
If you are going to be using the AVT system with a radio, you will also need to connect a cable from the female RCA jack on the AVT hardware that is marked **To Tx**, over to your radio. The actual point on your radio you will connect it to will vary, depending on your particular equipment. The AVT systems audio output is varied from zero to six volts peak to peak under software control. The six volt maximum level may be excessive for some radios, and you may wish to place a pair of resistors in circuit so that the maximum level of the output from the system is reduced. The following illustrations and instructions may be helpful.



2 resistor audio pad

In any case, you can connect this audio output signal to a phone patch input, the mic input, or possibly to a modem data input on your radio, *if* it is not a digital input. There is no output on this line unless the Amiga is specifically transmitting video or sending your station ID, so you can connect it in parallel with your microphone if that is most convenient. If you choose this type of hookup, find out if your microphone is "live" when the push-to-talk button is *not* depressed. This is because the AVT can key your radio all by itself, using its keying output lines, and if your microphone is "live" any time the radio is transmitting, you must take care not to make any noise during image transmission - even better, arrange some way for the microphone to be defeated when the AVT is transmitting. It is best to be aware of this situation from the first, if it exists.

The transmitter output RCA jack provides a ground reference on the shell, and the audio signal on the center conductor of the cable.

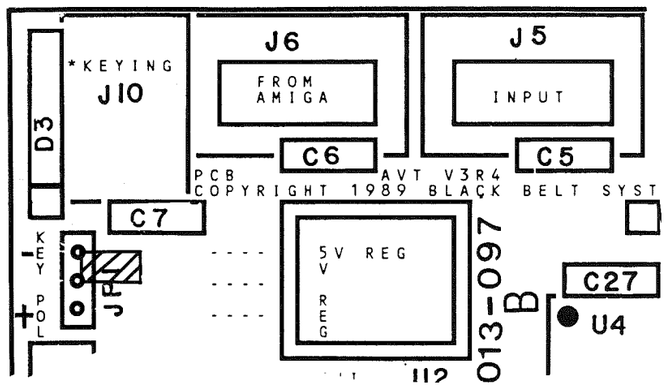


TX connection to radio

For those who are going to be using the system with a radio, you may wish to connect the AVT systems transmitter keying line to your radio. This line allows the Amiga to cause your radio to go directly into transmit mode just as if you had pressed the microphones push-to-talk bar.

The AVT system can key a radio transmitter in two ways - it can pull a nominally positive line *down* to ground, which is the usual means for a modern transmitter, or it can pull a nominally negative line *up* to ground, which would be appropriate for a tube type transmitter that uses a grid-blocking technique.

The AVT hardware is supplied set for the modern, positive-signal-pulled-to-ground type keying. If you will be using grid block (negative) keying, you'll have to remove the cover from the AVT hardware and move the **JP1** jumper block to another set of pins, as indicated in the following illustration.



Negative keying jumper hardware

To make the keying connection to your radio transmitter, there is a miniature phone jack provided on the AVT hardware. The center (probe tip) of the phone jack is the keying line, and the longer connection (probe shaft) of the phone jack is a ground reference.

You should also be able to wire this in parallel with your microphone, although you *must* check your radio and microphone circuitry to be certain. Some radios will experience an oscillation if a load is connected across the microphone audio signal input, and some have isolated ground circuitry.

If you're going to be using the AVT system over the telephone, then you simply connect a standard modular telephone cord to the RJ-11 female phone jack on the AVT hardware. You may parallel the AVT system with a standard telephone - it does not load the line appreciably unless it has been activated by you, and then it offers a load similar to a single telephone to the telephone line.

If you are going to be using the system with a typical amateur radio station, you may have more than one signal source. Tape decks are often used to save the older style SSTV frames, and many amateurs have more than one radio.

You can connect up to five different audio inputs to the AVT hardware on the input jacks marked **INPUT 1** through **INPUT 5.** To switch the transmit signal, you may need to provide a rotary switch of some type. However, you may find that you do not need to switch the transmit signal; since the signal will only go out over a radio that is keyed, if you utilize manual keying, you have complete control over which rig transmits. There are many ways to wire a multiple radio station, and this we leave to you. The components of the system are simple; one line that will *key* on transmit, and one line that contains the audio signal. It should be easy to set these up to meet your needs, whatever they are.

Once you have connected the appropriate audio cabling for your configuration, it's time to plug the AC adapter into the unit, and then power up your Amiga. We will now backup the software (this is *extremely* important!) and then you can begin operating the system.

Backing Up

Once your Amiga has been booted using your normal boot procedure to either the WorkBench or the CLI, utilize standard backup procedures to make a complete copy of the AVT Master diskette, or even two copies. Once this has been done, place the original in a safe place and *leave it there*. Only use the original diskette if the copy(s) you have made become unreadable or unusable in some fashion, and then only use it to make more copies.

If you utilized the WorkBench "Duplicate" menu item to create your copy, you'll also need to use the WorkBench "Rename" menu item to remove the text "Copy Of " from in front of the diskettes name. Note the space that comes after the word "Of"; when you have a disk that is named "Stuff" and you use the workbench "Duplicate" menu item to create a copy of the disk, the copy is named (by the WorkBench) "Copy Of Stuff". If you don't pay careful attention during the rename process, you can end up with a disk that is named " Stuff", which is a six character name that begins with a leading space... This can be extremely confusing, especially to the new user, when references to volume "Stuff" result in messages that indicate there is no such volume inserted in any disk drive -- the system is aware of a volume called " Stuff", but you didn't ask for that one, did you?

Software Installation

On the release disk, you will see two of the drawers (or directories, if you are examining the disk from the CLI or a utility dir program) are called **For_C** and **For_S**. These directories contain programs and datafiles that need to be copied to the named directories (C, and S) on your system disk (or partition, for hard drive users).

Two programs in the **For_C** directory need special attention here.

BackShow is a utility program developed for the AVT system that displays most any type of IFF image file. It has two unusual characteristics that make it especially appropriate for use with the AVT system, more so than any other "show" type program. The first of these is that of creating the image on a screen *behind* all the other screens. Since it is intended to display images that the AVT will "grab", it doesn't bring the image to the front at all. The second characteristic is that if an image has "color cycling" enabled, this utility recognizes that but does *not* cycle the image. This prevents the AVT from grabbing the image when the colors are in some intermediate state, which can happen using a more conventional show type program such as SuperView or uShow. Make sure you copy this to your C: directory.

NextScreen is a utility program that has been placed in the Public Domain by its author. We include it here because it is extremely useful when the AVT system is operating. We'd like to thank the author of NextScreen for creating this utility and making it available to the Amiga community. Its function is to allow you to bring the next screen to the front by pressing the left Amiga key and the M key, together. Normally, the **Left-Amiga-M** key combination brings the current screen to the front, and the **Left-Amiga-N** key brings the WorkBench to the front. With NextScreen installed, **Left-Amiga-N** still brings the WorkBench to the front, but **Left-Amiga-M** cycles from screen to screen. To use it, place the program in your C: directory, and place the following line in your startup-sequence file (the startup-sequence is located in the S: directory on your boot disk or partition):

```
run >nil: c:NextScreen
```

There are several types of files that the AVT system uses for various purposes. Each of these files has a file **extension** that is unique to that type of file. These files are kept in the same drawer or directory where the AVT_Master software is kept. You'll need to move files from the release disk to the directory where the AVT system will be operating if you do not plan to use the software from a direct copy of the release disk. The files that need to be moved have the following extensions:

.mcr - Contain sets of ten ARexx macros, general use
.curve - Contain ASCII demodulator definitions, FAX use
.cbar - Custom color bar definitions, SSTV use

One or more samples of each of these file types are on the release disk. We strongly suggest that you keep all of them with the AVT_Master software so that when you get around to experimenting with each of these items, there are examples for you to examine.

In addition to file types that the system uses to customize itself mentioned previously, it produces and/or uses files of various other types. Here is a complete list of all types of files handled by the AVT_Master program.

.low - 128x120 color SSTV files
.bwl - 128x120 B & W SSTV files
.high - 256x240 color SSTV files
.bwh - 256x240 B & W SSTV files
.avt - 320x200 color SSTV files
.Avhi - 320x400 color SSTV files, may be 3-D (Haitex)
.fax - 1024x1200 B & W FAX files
.iff - Amiga Interchange File Format image files
.curve - Contain ASCII demodulator definitions, FAX use
.mcr - Contains sets of 10 ARexx macros, general use
.cbar - Contain custom color bars, SSTV use
.rexx - Contain ASCII macros and scripts

Getting Started

Before you can effectively use your new AVT system, you must understand how to operate your Amiga computer. It is beyond the scope of this manual to teach you how to operate the computer itself. If your Amiga is new to you, please take some time to familiarize yourself with it's operation before attempting to operate the AVT system. This will eliminate many hours of frustration and confusion. The AVT system works in a manner similar to any good Amiga software, and so is easily learned by those familiar with the Amiga. Those who are not familiar with the Amiga will experience some degree of confusion because of a lack of basic operating information that relates to the entire computer system.

To start the AVT software operating, double click on the AVT_Master icon. After a pause to load the program into memory, a small window will appear containing copyright information and so on. A few moments later, the software should come up ready to operate. There are many configuration options which can be changed using the workbench **INFO** selection to edit the icon's *tooltypes*. Refer to the Amiga computer owners manual for information on editing ICON tooltypes.

If the Amiga appears to lock up, or presents a requester to you that indicates that the hardware is not operating correctly:

First, *turn off the power* to the Amiga.

Next, make certain that you have the AVT hardware connected to the parallel port, and if there is a switch on the parallel port, that the AVT hardware is the item selected in the current switch position. If this all checks out, make sure that the AVT hardware has power applied to it. If the hardware is not operating, the program will *not* start up in a fully operational mode.

When the AVT System begins operation, it comes up showing two screens. One is full sized and completely black, the other a short, full color screen containing the graphic control panel.

The full sized, black screen is used to display SSTV and FAX images to you when they have been received or loaded from disk.

The colored screen, which is the control panel, has a number of "buttons" that (usually) contain indicator LED simulations at the upper left of the button. Observe the LED simulation to see what the current status of that particular button is. In addition to these buttons, standard intuition menus are available in the control panel screen when you press the right mouse button.

Various functions are available from the menus. Some buttons select video modes - others open windows in the control panel that allow you to select other types of operations. For instance, one button has a broom and dustpan on it. This button opens a window with various *new* buttons in it that provide a range of tools that "clean up" other images.

There are a lot of buttons, and a lot of windows, too, so don't expect to figure the entire system out in one sitting- or even several. The thing is, you don't *need* to - basic operation of the AVT system is very simple indeed.

(SCV-400
Sony-
DA-Pro 4Head
1- Tuner=

To receive an image, you press the button that matches the mode of the image to be received. For instance, the top left button selects 12 second color composite, or eight second black and white (B&W). Point at the button drawn on the screen with the mouse pointer, and using the left mouse button, "press" it. The red indicator at the upper left of the button illuminates, and you are now in this mode. To see if you are in color composite mode, look at the button in the third row, first column; it says "CMP" on it, which is short for composite. If it is illuminated (red indicator showing) then you are. If not, point the mouse at it and press it like you did the 12/8 button. It illuminates, and you are in 12-second composite mode.

You can select any one of the 5 receiver inputs, or the telephone, by clicking on the **I/o** button and pressing the appropriate box gadget. Don't click on the little red indicators; use the boxes. If you have connected your signal to **Input 1** on the AVT hardware, you should not need to do this at this time unless you are using the telephone interface. If this is the case, click on the **I/o** button, and then select the **Telephone** box. Dont' do this until you have someone on the line however, as this does "pick up" the telephone, that is, take it off-hook.

Now, to receive an image, press the "RX" button (short for receive). The system begins scanning, and you're off and running!

Initially, to receive images you'll probably want to select AUTO mode by pressing the "AUTO" button. This mode causes the system to attempt to determine what kind of video is being transmitted automatically. If it recognizes the signal, it will automatically select the correct buttons and begin receiving. If it does not recognize the signal, it will not begin reception - this can happen if there is interference during the beginning of the video frame.

When you are in AUTO mode, the mouse pointer will not move- the system is running with its complete attention on the audio coming into the AVT hardware. To exit the AUTO mode, press the left mouse button once. This also applies to two other situations - the mouse pointer is locked during both receive and transmit, and to exit these modes, again, you simply press the left mouse button once.

When in AUTO mode, the AVT system will recognize and start reception of the following 26 SSTV types:

AVT 24, 90, 94 and 188 second color standard
AVT 24, 90, 94 and 188 second color QRM compensated
AVT 125 second B&W
Robot 8, 12, 24 and 36 second B&W
Robot 12, 24, 36 and 72 second Composite Color
Scotty S1 and S2 color
Martin M1 and M2 color
Volker-Wrasse 24, 48, and 96 second color
Volker-Wrasse 7, 16, and 32 second B&W

In addition, the AVT system sends the appropriate information in the vertical sync pulse at the beginning of each of these types of images so that the receiving system can differentiate the correct mode and start reception, if that is appropriate.

To transmit an image, generally, you select the mode (just as you did for receive operation) and press the TX (short for transmit) button. The only exceptions to this are the color composite modes. There are four of these, and in order to transmit in these modes you have to perform two extra steps. First, point the mouse at the button that is labelled "Synthesize" and click on it. A window will open, titled "Synthesis and Conversion". In this window, there are four buttons that correspond to the four composite mode buttons on the main control panel. Simply point at the button that matches the mode you want to use, press it using the mouse, and the system will prepare the image for transmit in just a few seconds.

Basic Operation

Let's begin with one important thing in mind - this is a complex system; it may have more capability than any other single communications tool you have ever operated previously, and because of this you should *not* expect to figure the entire system out in just a few sittings. The manual's basic operation portion is organized so as to present each *type* of operation in a separate section that covers that type of operation in a reasonably complete manner.

SSTV Operating Protocol

When operating SSTV on amateur radio, the first thing you need to be aware of, as with any amateur mode, is that you must comply with certain FCC rules.

* You must identify your station when you first come on the air using voice or CW (the AVT Master can do this for you).

* You must, as long as you remain on the air after your initial appearance, identify your station again using voice or CW, at *least* every ten minutes.

* You must **not** utilize amateur radio for the transmission of images that are going to be used in any commercial manner.

* You must **not** send "obscene" material using SSTV or FAX modes. This is a somewhat vague rule, and is subject to considerable interpretation. Use your best judgement - and keep in mind it may come down to a pink slip if your definition is looser than the FCC's... and this slip won't be the kind with lace on it!

* You must **not** willingly interfere with other amateur stations.

With these rules in mind, you're ready to begin with one of the most entertaining special purpose modes in amateur radio - SSTV!

As far as operator to operator protocol goes, it's really quite simple. First, ascertain if the other station can handle the mode you are planning to send - if not, determine what the best mode is they are able to utilize, and then change the image to that format and send it. There is little chance that the other station will have any modes that you cannot accommodate, as the AVT is currently the richest in different mode capabilities of any system on the market.

Since the AVT System (we'll just call it the AVT from here on) can do over 40 different SSTV modes, we're going to break them up into logical groups. This will make it easier to understand how the program is laid out, and also make it easier to utilize this manual as a guide during actual operation.

Images As The AVT Sees Them

To the AVT, images are kept in one of three internal formats for almost all SSTV operations. These are:

avt - 320 pixels across by 200 pixels down
high - 256 pixels across by 240 pixels down
low - 128 pixels across by 120 pixels down

The AVT also supports the following internal formats which are not compatible with the previous three:

avth - 320 pixels across by 400 pixels down
super - 640 pixels across by 400 pixels down

These two formats are also both 3-D image compatible, if you have the Haitex(tm) 3-D glasses; you can only send and receive images of these two types when working SSTV with another Amiga using an AVT system.

You need to be aware which of the avt, low, and high formats each SSTV mode utilizes. This is because these formats are completely different internally. There are functions which quickly and conveniently change one format to another, if this is required, but in order to use these functions you have to know both what mode you are in and what mode you want to convert to. Throughout this manual, as each SSTV mode is explained, the format that is utilized by that mode will be pointed out as required.

Try and keep the distinction between an AVT *format* and an SSTV mode clear in your mind:

An AVT ***format*** refers to the organization of an image in memory -

128 by 120 (low);
256 by 240 (high);
320 by 200 (AVT);
320 by 400 (AVT high);
640 by 400 (AVT super);

An SSTV ***mode*** refers to a particular method of sending an SSTV image -

Robot Composite;
AVT QRM;
Line Sequential Color;
Microcraft Hi-Res;

The terms *format* and *mode* will be used carefully and consistently throughout this manual.

About The RGB/Composite Main Memory

The system maintains all SSTV images internally in a memory with three components. Each color image can be thought of as consisting of three separate components - an image which contains all the red information, an image that contains all the green information and an image that contains all the blue information.

When the AVT displays a color image on the screen, it takes the contents of all three of these memories and combines them so that a "normal" color image is created. You can look at the red, green, or blue component of these memories by itself at any time by simply pressing the appropriate control button. If you are not sending color images, it is possible for each of the red, green, and blue memories to contain a completely different black and white image. If you ask the AVT to display a color image under these circumstances, it will try - and you will get a ghostly overlay of your three black and white images in three primary colors. It's interesting, but not particularly useful.

Saving And Loading SSTV Images

The AVT offers you disk-based storage of any SSTV image format. Note that the images are not stored with any relationship to the mode you received them in, instead they are stored according to the internal format - low, high, AVT, AVT high or AVT super.

This means that on disk, an image that was received in AVT QRM 24-second mode is no different than an image that was received in Robot 12-second composite - both are low format images.

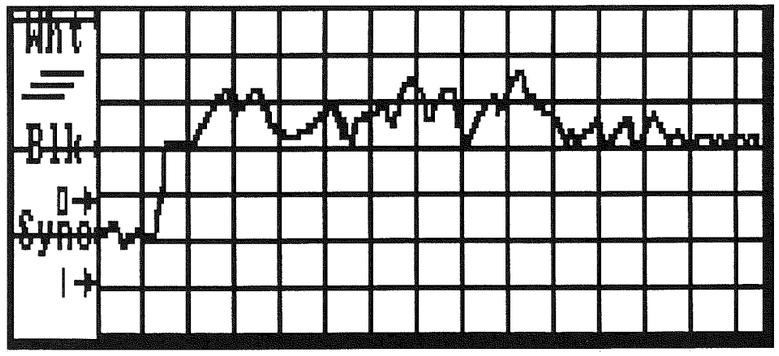
The most important thing to keep in mind when you want to save an SSTV image is that the AVT system takes note of what SSTV mode you have selected when you save something, and chooses the appropriate format. Put another way, if you receive something in 12-second Robot composite, then simply make sure that mode is also selected *before* you choose **SAVE** from the SSTV menu. This way, the AVT will always save the correct type of image.

Once you select the **SAVE** menu item from the **SSTV** menu, a file requester is presented to you. This file requester uses standard Amiga concepts and tools to assist you in moving around the Amiga's file system. You can choose new disks, new directories, and so on. You will see an area marked **extension**. Here (if you are saving a low format image) you will see an extension that reads **.low** in the box. The image file will be saved using this extension.

In order to save a file, you must first give the file a name. There is also a box for this, marked **Filename**. Enter the name you wish to use (Girl, Bird, CallSign, etc) and press the **enter** key. Now press the button at the upper left of the file requester marked **SAVE**. When the AVT has completed saving the file, you will be returned to the AVT control panel. You may follow standard Amiga file name conventions here - spaces, letters and numbers are all allowed, in any position in the filename.

Using The Tuning Scope

The AVT provides you with an accurate tuning aid configured as an oscilloscope. In order to use it, you need to know what a typical slow scan signal looks like:



Scope with synced signal on it

That signal is typical of a Robot composite signal, or one of the other SSTV modes that use horizontal sync. The sync signal is the one that appears at the left of the scope, and extends the furthest towards the bottom of the scope display. Notice the calibration marks to the left of the display graticule. One is marked **sync**, while two others are marked for black and white.

To tune any SSTV signal that has sync on it, simply align the lowest sync tip with the **SYNC** marking on the scope. Assuming that the transmitting station's equipment is in alignment, you are now perfectly tuned.

Tuning a signal that does not have sync on it, such as FAX or the AVT modes is more difficult. Basically, you tune your radio so that the signal does not extend past the black and white markings on any brightness peak.

When working another AVT system, do not use an AVT mode signal to attempt to tune in the other station. Instead, request that the other AVT station send you an *alignment tone*. Likewise, should another AVT station wish to tune to your signal, open the **SET** window and press the **Align** button. This sends a pure black line with sync on it. To tune using the alignment signal, adjust your radio such that the sync tip is on the line marked sync, and the rest of the signal aligns with the **black** indicator.

Another useful function that the scope performs is to show you how much time is left when you are transmitting a slow scan signal. A vertical bar graph fills from the bottom towards the top as the image goes out.

Detailed Information About The Scope

The scope is affected by a number of the buttons on the main control panel. The button marked **Filt** activates a digital lowpass filter that affects both the scope and the received signal. When on, the effect of noise on the received signal (and the scope display) is reduced.

The **Trig** button turns the scope triggering on and off. This scope will retrigger whenever a signal extends down into the sync region whenever this button is on. If it is not on, the scope free runs at the current sweep rate.

The scope's sweep rate is changed whenever you select a particular SSTV mode from the main control panel. Sweep rate is adjusted so that approximately one entire scan line will fit on the scope. When using the scope for generally nosing around the shortwave bands, select different SSTV modes to change the sweep rate. For instance, selecting the **12/8** button and the **Grn** button will select the fastest sweep rate. Selecting the **36/72** button and the **CMP** button will select the slowest sweep rate.

Some operations will cause the scope display to be erased. If this occurs, simply point the mouse at the scope display itself, and click the left button once. This causes the scope display to be redrawn with the last information on it.

Things That Happen At Startup

The AVT system does a number of things for you when it starts up.

If you have started the program from the WorkBench, then it will look "inside" the icon for information in the ToolTypes. Here, it finds all the startup parameters you set, or the ones that were placed there as the default values by us. If you'd like to see what you can change, please refer to **Appendix I** for explanations of each of the many options available to you.

If you started the program from the CLI, then the options that are selectable on startup are controlled by command line parameters. Again, you can get details on these by referencing **Appendix I**.

Note: We would like to take the time to tell you about one particular option - the **chip** option. If you are an a500 or a2000 user with only one megabyte of memory, you *must* use the **chip** option when starting the program. If you do not, many of the receive operations will not function correctly. Please refer to **Appendix I** !!!

If you have ARexx installed in your system (and we certainly hope you do!) then the AVT Master software will attempt to load the macro definition file "Default.mcr". This is a file of 10 macros that initially we have set up for you, but you can change, of course. Since the Default.mcr is loaded automatically upon startup, these macros are always available to you. The Default.mcr file supplied on the release disk has some interesting items in it - we think you'll enjoy exploring them. Take special note of the **F1** macro; This invokes the ARexx script "rexx:filegrab.rexx", which allows you to easily grab IFF images using a file requester. The **F2** and **F3** macros are a "matched pair" of macros. **F3** is used to transmit an image over the telephone, and **F2** is used to automatically receive an image as sent by an **F3** macro.

If you *don't* have ARexx installed, the program will notice this and tell you about it. This is so that you are aware that you don't have any macro or script capability available.

The AVT Master also attempts to load the file "demod.curve". This is the default demodulator profile for the FAX receive mode. It allows you to define the linear characteristics of the FAX demodulator. As supplied from us, it's just a normal linear curve. If you like, you can change it to a log curve, or any arbitrary curve.

The AVT Master software checks to see if the AVT hardware is attached to the computer. If not, it warns you about it, and then lets you proceed. You only need the hardware if you're going to receive, transmit, use the scope, or the macro functions that read the period from the hardware. You can manipulate files, look at pictures, write text on them, and so on, without the hardware attached. Just don't use the rx, tx, or scope functions and you should have no problems.

If the hardware is attached, using information present in the ICON or from the command line in the CLI, the program selects one of the 5 receiver inputs on the hardware. It can also pick up the telephone, but you will probably not want it to do this immediately upon starting up. The option is there if you need it, though.

Once the program is satisfied that all is well, it will fade the WorkBench away and bring up the lights on the AVT Master control panel - and you're ready to go!

Special Section For Users With One Megabyte Machines

Although the AVT system will operate in machines with only one megabyte of memory, some things work differently, and certain operations are not available at all. The following sums up the differences for you:

1) You *must* use the **chip** keyword when starting the AVT Master software from the CLI, or, you *must* have the **chip** keyword in the "PARMS=" line in the icon. Refer to **Appendix I** for exact details on program parameters.

2) The program maintains an extra display screen that is black and white *only*. This screen is kept "on top" of the color screen during most receive and transmit operations, and is either removed automatically or can be pushed away by you when you are not doing one of those two things. The reason for this is that when the six bitplane hold and modify screen is showing, a one megabyte Amiga has to slow down the processor to almost half speed; and under this condition, it is not possible for the software to get all the work it has to do done in time. With a four bitplane "normal" screen showing, the CPU runs along quite nicely. You will find that if you are not running the NextScreen program contained in the "For_C" directory on the release disk things will be inconvenient.

3) You will only be able to have one main memory. Each additional image memory takes almost 200k; you can have four low resolution or one high resolution image in that one main memory.

4) You will not be able to run the AVT 188 (320x400) mode at all, as you don't have enough memory.

5) FAX operation will be limited to 640x400 images. When you have about 1.5 megabytes of memory or more, this expands to 1024x1200.

6) The FAX Control window will not be available. This means that you will not be able to rotate, flip, threshold, or pan around FAX images. Again, this changes when you reach about 1.5 megabytes of memory.

7) You will generally be unable to run much else besides the AVT system, simply because it doesn't leave enough memory free in a one megabyte machine.

8) We strongly recommend that you do *not* use the Voice ID feature at all in a one megabyte machine. It will work, but loading the Narrator and Translator libraries costs you quite a bit of memory, which you may need for other things.

9) Loading fonts also costs memory - don't open more than one or two during the operation of the system.

About the "Fatter Agnus" Chip

The "Fatter Agnus" chip allows your a500, a2000 or a2500 machine to have one megabyte of chip RAM instead of 512k. This is an ideal situation for the AVT system, which uses about 300k of chip RAM all by itself. When you have another screen open, for instance when you are grabbing images from a paint program, chip memory becomes a *very* scarce resource.

We do recommend that if at all possible, users with older machines obtain the Fatter Agnus upgrade from their Amiga dealer.

There is a project currently under way that looks like it will provide a means for a1000 users to utilize the Fatter Agnus as well. For the latest details on this project, contact Greg Tibbs on the CompuServe Information Service network in the AmigaTech forum.

Things To Watch Out For

Low Memory

If you have little memory (in the range of 100k or so) left, you should be aware that the AVT system may not have enough "headroom" left to operate properly in all cases. You may experience system crashes or lockups in this situation. You must either reduce your memory usage, or purchase more memory.

Disk Changes

You must *never* change the disk in any of the Amiga's floppy drives when the mouse pointer is locked. Examples of these times are when you have pressed the TX, RX, or AUTO buttons. The Amiga is "asleep" at this point, and is not watching what you are doing. When you remove the disk, the system loses track of what is in a drive and what isn't. If you then put in a different disk, the system will crash completely.

Inability To Get From Screen To Screen

This is usually a result of not having installed the **NextScreen** program as recommended here. You should be able to get to the next screen at any time by pressing the left Amiga key and the M key at the same time.

Menu Selections To Do With Macros Are "Ghosted"

This is because ARexx is not installed in your system. ARexx is currently an extra-cost item, available from your dealer or Black Belt Systems. When the 1.4 revision of WorkBench comes out, ARexx will be included with it for free. Of course, the upgrade to 1.4 will not likely be free. Once you install ARexx, the menu selections and the function keys will be available for your use.

FAX Control Button Is "Ghosted"

This is because you didn't have enough memory available to store the received FAX image. This requires 614,400 bytes of free memory, in four 153,600 byte "chunks". The only solution is to get more memory. Systems with 1.5 megabytes (or more) memory generally will have enough memory to store complete FAX images. When the system cannot get enough memory for a complete FAX image, it stores the image in the actual screen memory. This works fine, but has two drawbacks. First, the screen is only 640 by 400, so that is the maximum resolution of FAX image you can have. Second, this memory is *chip* memory, which is slower than other memory that may be available in the system and so slows down the program when it has to do anything with the actual FAX data.

Synchronous Operation

In the SSTV menu you will find an entry marked SYNCHRO. When you select this item, you are presented with a series a "trimmer values". These values are broken into two groups. On the left are four values and a button which are used to operate the Robot composite modes in a synchronous manner. The default is that these modes use the sync pulses for vertical alignment; if you press the button next to the four leftmost trim values, it will switch back and forth between "Use Sync" and "Synchro" positions. This is an advanced capability of the AVT system.

The values in these boxes may require adjustment for your particular Amiga; that is why the window is available. In general, you should not attempt to use synchronous receive on the Robot modes except when working another AVT system.

On the right hand side are eight more trimmer values. These are used in conjunction with the Martin and Scotty S1, S2, M1 and M2 modes. In contrast to the Robot modes, these modes are *normally* synchronous. The default here for the left SYNCHRO button is the "synchro" position. You can select the "Use Sync" position, but you will not get as good a resulting image. You may wish to do this, however, if you are trying to receive an S or M mode image from tape.

The default values in all of these gadgets were determined using an Amiga a1000 with two megabytes of extended memory. If you have a different setup, you may find that these values need to be modified. If so, use the text at the top of the trimmer window for a guide as to how to adjust these numbers.

Once you determine the correct settings for these trimmers, you can set them by utilizing the SET commands in the programs icon. See Appendix I for details on these icon setup parameters.

You can also utilize the ARexx SET commands to switch between various settings for these values *if* you have ARexx installed. You may wish to do this because Robot composite modes are sent at slightly differing speeds depending on which particular piece of equipment is doing the sending. For instance, the following units all send at different speeds:

Robot 1200c
Robot 450c
Robot 400c
Robot 1200c with Scotty ROMs
Robot 1200c with Scotty ROMs and his Oscillator kit
Robot 1200c with Martins ROMs
Robot 1200c with Martins ROMs and his oscillator kit
AVT System

In addition to this, individual "stock" Robot 1200c, 450c and 400c units send at various differing rates. This is due to the use of a very loose tolerance crystal oscillator for the control signals.

You will probably find that the synchronous settings are of most use when used with the AVT system, and the 1200c when using either Martins or Scottys precision oscillator. If you are planning to use the AVT mainly over the telephone lines with other AVT users, the synchronous modes will work extremely well for you.

ROBOT(tm) Compatible Operation

Brief History

Robot Research Inc. has been involved in SSTV for many years, producing some of the first commercial SSTV scan converters, as well as some easily-operated modern units recently.

Robot composite modes were developed to meet several disparate goals. These goals included single frame color, like the line sequential modes; but the new Robot composite modes were designed to avoid the problem of out of color sync images, and do so quite successfully. In addition, the frame time of images at any particular resolution was to be reduced - 24 seconds for a low resolution image was cut down to only 12 seconds, and according to Robot corporation, without a great deal of loss of picture information. Finally, an important goal for various reasons was to make these new color modes compatible with the old eight-second black and white standard as well. In all of these endeavors, Robot succeeded to some degree.

Robot composite color modes break an image into two components. Each scan line of an image is sent broken into these, and reassembled at the receiver into a full color scan line.

The two components are called *luminance* and *chromanance*. Luminance is essentially the brightness of the image, without the color information. Chromanance is the color information (the HUE) without the brightness information. Technically, anyway. In practice, it's not quite so, but it's close enough to be discussed in these terms.

Robot 12-second color sends exactly the same amount of luminance that an eight second black and white image does, then the chroma after that. The result of this is that an older eight second black and white unit traces the luminance normally, and then "hangs" at the end of the line waiting for the sync pulse. As a result, the chroma information is essentially ignored by the receiving black and white unit, unless noise causes the receiver to trigger accidentally. This is the basis for the compatibility between the 12-second color and the eight second black and white modes.

Robot 12- and 36-second color modes are similar; each sends a line of luminance for each line in the image, and each sends 1/2 a line of chromanance for every *two* lines in the image. This results in a received picture with all the luma (brightness) information, and 1/4 of the total color (chroma) information. For images that consists of smoothly changing color contours, this is sufficient, and results in a surprisingly good image. For an image that contains highly detailed color information, however, the result is poor to terrible.

To address this, Robot added the 24-second and 72-second composite modes. These modes send a line of luminance for each line of the original picture, and they send 1/2 a line of chromanance for every line in the picture.

The result of this is much better encoding for images that have highly detailed color information that is detailed *only* in the vertical aspect, since each scan line does get its own vertical color information, instead of an average of two lines, like the 12- and 36-second modes. However, if the image has highly detailed color information in the horizontal aspect, that is, in each scan line, then the image again can tend to be quite poor.

These new Robot image transceivers (scan converters) can still send in the RGB frame sequential modes, so for highly detailed images one should use the RGB 111 types of transmissions. Unfortunately, this means that the advantages of the composite modes - single frame color, short total frame times - are lost.

In addition to the rather poor color resolution offered by the Robot composite modes, there is one specific disadvantage brought on by the composite encoding method. For a Robot composite image, what chroma detail an image is to have is determined by a tone at a particular frequency. If the received image is tuned off frequency, as is common when using SSB radio transmission, the image will "tint" itself red or green, depending on the tuning offset. Only when the image is tuned exactly on frequency will it be received in the correct colors.

To correct for this tendency, Robot designed the scan converters to utilize the starting sync pulses to "auto-correct" the received image - the AVT uses this technique also - and so *if* the scan converter is in AUTO mode (likewise the AVT system) the image can be up to a hundred cycles or so off in either direction and the colors are still somewhat correct.

During manual reception, this is not done, so color tint problems are common when operating the Robot modes.

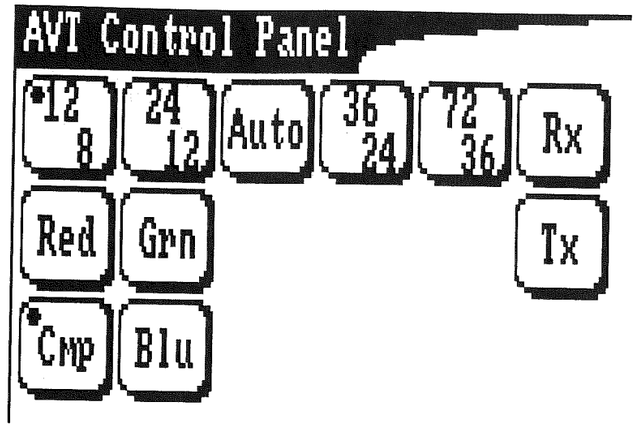
The justification for reducing the chroma information in a Robot image was given as what is essentially a quote from the NTSC composite television receiver specification used for commercial television, determined by research done decades ago. This research determined that the human eye was less sensitive to color information than it was to luma information. As a result, they designed the color television signal to be heavier on the luma information than the chroma. All in all, it works fairly well - on a television.

For SSTV, however, the same reasoning can be shown to be false, both theoretically and in actual practice. The problem here is that the *resolution* of a color SSTV image is so low compared to what the human eye can discern, and a color television, that color artifacts due the luma/chroma information imbalance are immediately apparent even to the untrained eye. The original reasoning was postulating a system where the chroma information would provide about 100 color changes per line - but the encoding method allowed these changes to occur *anywhere* on that line - not at 100 evenly spaced locations, such as is the case with Robot's composite SSTV modes. In addition, a color television provides about 280 visible luminance changes in a line along with these color changes, so the low resolution composite SSTV is less than 1/4 the luma, and 1/16th the chroma information... lastly, a color television changes the screen disply 60 times per second, and this constant update of the display creates a much more dynamic visual environment for color changes, the result of which is simply that you usually don't see too much degradation.

Despite these rather glaring shortcomings, Robot composite SSTV has become the most commonly transmitted SSTV mode sent on the Ham bands. This has been due to a complete lack of qualified competition... until now, that is.

Operation

A large amount of the SSTV you may hear on the bands at one time or another will be in what are called the "Robot modes". These include four black and white image modes, and four color image modes. A number of the buttons on the AVT's control panel are used to control these modes. They are detailed in the following illustration:



Auto, cmp, r, g, b, 12, 24, 36, 72, rx, & tx buttons

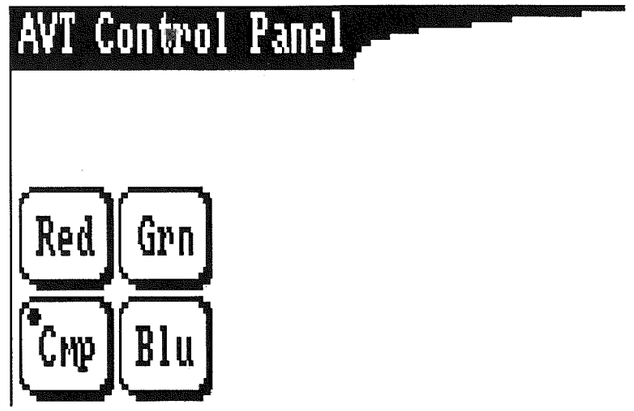
The **AUTO** button is useful when receiving Robot transmissions, as all eight types send a "VIS" code which, if you are in AUTO mode, will cause the AVT to automatically begin reception at the correct time.

When you want to operate Robot modes manually, you need to know what mode the other station is going to be sending. How? Ask them, if they don't volunteer the information themselves. Without this information, it can be difficult to get things set up correctly.

Just as an interesting point, after awhile, you *may* be able to operate manually without being told what modes are being sent - the trick is in training your ear. Since these are all modes which utilize audio frequencies you can easily hear, and since the modes are quite different in duration and encoding method, it is usually possible to tell within just a second of two after an image has started exactly which mode it is. Some individuals will develop this capability immediately.

As mentioned previously, there are eight different "new" Robot modes. Four of these are color composite modes, and four are black and white only modes. If you examine the four speed control buttons (the two to either side of the **AUTO** button) you will see that each speed button has two speeds placed upon it. One speed is in color, and one is grey. The one that is in color, naturally enough, is the speed that is utilized for the four color composite modes. The one in grey is used for the four black and white modes. Press the speed button that matches the mode the other station will be sending.

In addition to selecting the speed, you will need to press one of the four image memory control buttons detailed in the following illustration:



R, G, B, Cmp buttons

If you press the **CMP** (composite) button, this tells the AVT that you want to operate in a color composite mode, not a black and white mode. If you press any one of the **R, G** or **B** buttons, this tells the AVT two things - first, that you want to operate in a black and white only mode, and that you want the black and white image to be placed in that particular color memory.

Once these two steps are done, wait until the other station is about to transmit, then press the **RX** button, and the AVT will begin to attempt to receive. When the other station's transmission actually begins, you (hopefully) will see the AVT reposition itself to the upper left and begin to render the image from the top left to the bottom right in black and white - even if it's a color image.

Once the image has been received, if it's in black and white, you are done. If the image was a Robot color composite image, then the AVT has some more work to do. If you observe the screen carefully, you may be able to see a small white cursor travel down the left side of the black and white image that was displayed when the image came in. The AVT is *decoding* the color information. When this occurs, the image is being broken out from its composite format and being written into the red, green and blue memories. Once this completes (indicated by the cursor reaching the bottom of the image) the image will be redrawn for you in full color.

NOTE:

If you press the left mouse button while the AVT is decoding the color information, the decode operation will be aborted and the picture will be lost. This may be convenient if the image is of low quality and you have decided you do not wish to wait through the decode process, but you need to be aware that you can lose an image with an inadvertent mouse click here!

You should also be aware that any of the four speeds on the **8/12** and **12/24** buttons select the *low* (128 by 120) internal format, and any of the four speeds on the **36/24** and **72/36** buttons select the *high* (256 by 240) internal format.

Transmission Of Robot Mode Images

To transmit an image in one of the robot black and white modes, all you need to do is select the speed, which of the red, green, or blue memories the image is to come from, and press the **TX** button. The indicator on the button will illuminate for the duration of the transmission, and goes dark when the operation is complete.

To transmit an image in one of the Robot color composite formats, you proceed slightly differently.

First, click on the button that is labelled **Synthesize.** A new window will open, entitled **Synthesis and Conversion.** In this window you will see four "SYNTH" buttons, each bearing a particular composite speed indicator. Press the button that corresponds to the format you wish to use to transmit. A cursor will travel down the left side of the image on the screen (this indicates the image is being prepared for transmission) and then the window will close.

At this point, you will note that the correct modes (the speed, and the composite button) have been selected for you, and you can now press the **TX** button at your convenience.

Remember that the images you are dealing with in the Robot modes may be in either high or low format. If you receive an image in 12-second composite, this is a low format image. Perhaps you may wish to retransmit this image - if so, unless you utilize one of the format conversion functions in the Synthesis window, you must use a mode that is also low format - 24-second composite is your only other choice that is a Robot mode.

For example, to change an image that was received in 12 second composite color (which is low format) to a 36 second color composite mode image, you would do the following:

a) Press the **Synthesis** button
b) Press the **>256** button in the Synthesis window
c) Press the **SYNTH 36** button in the Synthesis window

This image is now ready to transmit as Robot 36-second composite color. It is also now in the **high** format. The **>256** conversion operation actually creates new pixels to reformat the image from 128 by 120 to 256 by 240. Special image processing techniques are used to ensure that the resulting image is not blocky, or "pixelized".

Tinting

The AVT is capable of easily tinting any Robot composite color mode image that has just been received, or just prepared for transmit using the SYNTH buttons in the Synthesis and Conversion window.

Because of shortcomings in the design of Robot's composite format, it is possible, even likely, that the tint of the image you receive may be too green or too red. If so, you can correct this with the **Post-Receive Tint** control in the Synthesis and Conversions window. This control has a knob in the middle, with two large open areas to either side of the knob. The ends of these areas are marked "Red" and "Green". To tint an image, place the mouse pointer over the empty area *beside* the knob - not on it - and click the mouse button once. After a delay for processing, the image will be redrawn with the new tint.

Keep in mind that this process **only** works on images that:

a) have just been received in a composite mode, or...
b) have just been prepared for transmit using SYNTH buttons

Format Conversions - Low To High, High To Low

Conversion between formats are easily done, if needed. Press the **Synthesis** button, and the **Synthesis and Conversions** window opens. If you have a low format image (Robot 12- or 24-second composite) and you want to convert it to high format (36- or 72-second modes), press the **>256** button. The ">256" indicates that the conversion operates to change an image's resolution *up* to 256 pixels per line.

If you have a high format image (Robot 36- or 72-second composite) and you want to convert down to a low format image, press the **<128** button. The "<128" indicates that the conversion operates to change an images resolution *down* to 128 pixels per line.

Saving IFF format images

The AVT system can save the image you see on the screen in a number of different IFF formats. If you are in one of the standard 256x240 (or less) resolution modes, the screen is saved as a vertical overscan 320x240 HAM IFF image. If you are in 94 AVT, the screen will be saved as a 320x200 HAM IFF image (the Amiga standard). If you are in 188 AVT, the screen is saved as a 320x400 HAM IFF image. If you are in 125 AVT or one of the FAX modes, the screen is saved as 640x400 four bitplane IFF image, *unless* you are in a FAX mode and have the **Fat Fax** gadget in the fax control window selected; in this case, the screen is saved as a 320x400 four bitplane image.

The New AVT Modes

Brief History

The AVT system offers several new, extremely high-performance SSTV modes. These modes have been designed to overcome or eliminate problems that have plagued conventional SSTV modes for years. You will find that AVT mode images can get through the most persistent QRM and QRN when other modes just show noise.

The AVT modes are the SSTV modes of choice - when working another Amiga station, you will almost always wish to use an AVT mode, since they provide much higher quality images than the composite, line sequential, or frame sequential type modes are capable of.

AVT modes are available that are of an equivalent format with any previously existing commercial mode. AVT 24-second mode is a low format SSTV mode. It may used to transfer images that were received using Robot 12 or 24, or 25.5 line sequential - in short, any SSTV mode that is a **low** format type. AVT 90-second is a **high** format SSTV mode. It may be used to transfer images that were received using Robot 36-, 72- or 51-second line sequential - any SSTV mode that is a **high** format type. There are many modes that are compatible with AVT 24 and AVT 90, for a specific correspondence, see **Appendix F.**

There are four AVT modes that have internal formats that are unique to the AVT system, and the Amiga. In other words, there is no corresponding SSTV mode in any other commercial or experimental unit at the present time. These modes, in general, reflect some graphics mode of the Amiga computer system that is beyond the capabilities of the standard SSTV resolutions. For this reason, you can only run them when you are working another AVT system - Robot systems and other similar units are "stuck" at the older, lower resolutions.

Operation

To receive an AVT mode image, you may press the AUTO button and let the system do the work for you. If signals are weak, you may find that the AUTO mode operation is unreliable. If this is the case, operate in the manual mode. To do this, select the AVT mode you wish to use by pressing the **AVT** button, and the appropriate speed button. AVT speeds are drawn in color, and the colors are *slanted* through the numbers - this makes it easy to tell which of the buttons are AVT speeds, as opposed to other types of speed controls. Once the speed and the AVT button have been selected, just press the **RX** button. The system will then wait until it hears an AVT signal come in, or until you press the left mouse button to abort the receive operation.

Operating 3-D With The Haitex Glasses

The Amiga offers you the unique opportunity to participate in high quality *stereo* (3-D) image transfer and viewing. Haitex corporation has created a high tech pair of glasses that, when connected to the Amiga, can cause one image to be applied to your left eye, and a completely different image to be applied to your right eye.

When you view things in the "real" world, your left eye always sees a different image than your right, for the simple reason that it's not in the same place, and the light rays that reach each eye come from a different angle.

With the Amiga, pairs of images can be created using paint programs, ray trace generators, and even games such that (when using these special glasses) an image with astounding three dimensional depth is created.

To operate with the 3-D glasses, the Amiga needs to be in **interlace** mode. This means that you must be running either AVT 188 which is 320x400, or AVT 125, which is 640x400. The glasses operate by using an LCD (Liquid Crystal Device) shutter on each lens, and the computer switches lenses at the same rate that the display changes images.

Haitex provides a utility program called **On3D** which is used to see the images. When you receive one, operate this program and put on the glasses. You'll be astonished at how good these images look!

Transmission

To transmit any AVT image, press the **AVT** button, the appropriate speed, and the **TX** button. That's all there is to it.

QRM Mode

AVT modes can be transmitted normally, or with QRM mode on. QRM mode doesn't increase the transmission time or the bandwidth, but it *does* increase the receivers ability to recover the image once the transmission has completed in a situation where there is heavy interference. For telephone operation, using QRM mode is unlikely to make much difference; however, when operating the radio, it can make a *lot* of difference! Once an image has been received in QRM mode, correction of the image, if necessary, is done with the **Line Correction** and **Frame Correction** tool sets.

You can turn the QRM mode on by opening the **SET** window and pressing the button there marked **QRM.**

For reception, AUTO mode will automatically notice and handle the QRM type transmission. If you are in manual receive, you should select the QRM mode in the **SET** window before you receive the image, or it will not be drawn correctly.

If you are in the wrong mode of manual receive, you can utilize two tools that are in the **Frame Correction** window. These are at the lower right and lower left hand corners of the window. Click on the icon that has the image that represents the type of problem the received image has. Either it will look like every other line is disturbed, or it will look like two half-screen images. The buttons are drawn in the window so that they kind of look like that - one button alternates every other line, the other is drawn in two short halves.

Narrow Mode

AVT modes can also be transmitted in a **narrow** mode. When narrow is in effect, a transmission uses the frequencies between 1,700 Hz and 2,100 Hz instead of the broader "normal" range which is 1,500 Hz through 2,300 Hz. Again, this is something that is unlikely to be of any advantage when operating on the telephone, but can offer substantial gains in performance for single sideband radio operation. You can turn narrow mode on by pressing the button in the **SET** window marked **Narr.**

This is dependent upon some features of the radio in use for reception of the images. The situation that must be taken advantage of is that the received image is literally narrower than usual for SSTV or FAX. If your receiver has passband tuning, IF shift, or IF slope tuning, then your receiver will be able to get the most advantage from narrow mode. The idea here is to reduce the bandwidth of the receiver such that the image still gets through, but interference on the sides does not.

In order to find out what is the optimum setup of your receiver, have someone send you a narrow-mode AVT image; it doesn't matter which resolution for this setup proceedure. While the image is coming in at your station (we assume you are using a radio here) observe the tuning scope and vary your bandwidth controls. See that the incoming waveform does not get severely rounded edges which indicates too narrow bandwidth for high frequencies. Also ensure that the signal does not break up on portions that extend towards the bottom of the scope; these are the low frequencies. We suggest you perform this proceedure under circumstances where there is good reception. It may be difficult to determine what adjustments for your rado are optimum if you cannot resolve the incoming signal fully.

A narrow mode AVT image is only 400 Hz wide. This is narrower than many CW filters. If you have an IF shift control, you can engage the CW mode and use the IF shift to move the narrower IF bandpass up to a center frequency of 1,900 Hz; this kind of operation is optimum for narrow AVT. The steep slopes of the crystal CW filter provide high immunity from interference on the sides of the incoming signal. Keep in mind that in order for this to work properly, the bandwidth of the CW filter in your radio must be 400 Hz or more. Really narrow CW filters, such as a 250 Hz filter, will not allow all the image components to pass through, resulting in a degraded image.

Format Conversions

Remember about the formats that are in use - low and high. You can convert an AVT 90 image, which is a high format image, to an AVT 24 image, by opening the **Synthesis** window and pressing the **<128** button. When the operation completes, the red indicator on the button will go out. At that point, close the window, and you're ready to change to AVT 24 and go to transmit.

Working SSTV In RGB (One, One And One Frame Sequential) Modes

Brief History

When SSTV was first introduced, images were sent in 8.5 seconds, and were black and white. The first color SSTV images were sent by using three separate SSTV scan converters, each capable of holding one black and white frame. The video outputs of these scan converters was connected to an RGB monitor, and each scan converter was then loaded with the red, green, and blue frames for the color image. This was done by using a black and white camera, and three appropriately colored filters.

Once the three scan converters were loaded in this manner, the image could be sent, one frame at a time, to a receiving station, which had a similar setup. As each frame was received into the appropriate scan converter, the image on the receiving station's RGB monitor would show the result, and after all three frames were in, the result was a full color image.

It did not matter what order the frames were sent in, as long as all three were sent, and they got into the correct scan converter at the receiving end. After a short while, it became standard to send first the red, the green, and finally the blue frames; the reason for this is simply the abbreviation - RGB - which makes the order easy to remember.

Operation

The AVT system uses the RGB order for an automated sequence of transmit frames. If you need to send the frames in another order for some reason, you just do it manually.

To send RGB, it is necessary to select the speed you wish to use from the black and white frame rates. These include 8, 8.5, 12, 17a, 17b, 24, 34, and 36 seconds. To select one of these frame rates, examine the buttons on the main control panel. The black and white frame rates are drawn in a single shade of grey, so that you can easily distinguish them from the color frame rates.

Once you have chosen the speed you wish to operate at, you then select the **111** button, directly under the **AUTO** button, and then press the **TX** button. The AVT system will send the frames one at a time, with a pause between each frame to give the receiving operator or system time to get ready for the next frame.

To receive an image in RGB 111, the method is very similar. Select the frame rate, press the **111** button, and then the **RX** button. Once the transmitting station begins sending, the AVT will automatically sequence from frame to frame as long as the signal is of good enough quality to receive.

There is one thing you can do during RGB receive that comes in handy from time to time; when you first press the **RX** button, the station at the other end may not be ready. In fact, they may talk for a minute or so, describing the picture. If this occurs, you need to be able to tell the system to wait. You do this by clicking the left mouse button once, quickly, before the scan has reached half the frame length. If you do this, the scan will begin again at the top of the frame. If you do it after the scan has passed half the frame length, then the scan will begin at the beginning of the *next* frame.

If you wish to exit RGB receive, then press the left mouse button and *hold* it until the busy pointer has changed from flashing red to black in the center. This is also true for black and white receive.

Line Sequential Modes

Brief History

After the early SSTV systems had been using frame sequential color for a while, hams began searching for a way to send single frame color. The objective was for the image to be received in color as it came in, instead of coming in and looking strange as the three color components combined at eight second intervals to finally form the full color shot after 24 seconds of signal.

To achieve this, the method developed was called *line sequential color*, as opposed to frame sequential color, discussed previously. As you may have guessed, line sequential color is similar to frame sequential, but instead of sending the R, G and B frames independently, each *scan line* was sent R, G and B. The result of this is that all the information for each scan line is present (full color) after just three scan lines. Hardware scan converters could utilize this to present the image in color as it scanned down the monitor, instead of after the entire transmission time.

Line sequential modes are now considered obsolete technically, as they have a strong tendency to confuse the color information in an image due to a missing scan line. If the receiving station missed the green line, for instance, then it will put the blue information in the green buffer, then the red in the blue, and so on. This results in a picture that has totally incorrect colors. This state is called "out of color sync."

The European version of line sequential color avoids this, for the most part, by sending a full intensity pulse at the beginning of the red line - it won't start the scan unless this pulse is present, and so avoids most out of color sync conditions. However, this mode is prone to losing lines if it misses that red burst.

Operation

To operate in a line sequential mode, you select the correct line sequential mode from the appropriate button. In the US, there have been three modes developed for this; 25.5-, 51- and 102-second RGB line sequential. These modes have their own buttons, located near the bottom left of the control panel. Once the mode is selected, press **TX** or **RX** to transmit or receive an image, respectively.

In Europe, the Volker-Wrasse company manufactures scan converters that also use line sequential techniques, but at differing scan rates. These rates are 24, 48 and 96 seconds. To set the AVT system to use these rates, you open the **SET** window, and press the **V-W** button. Close the set window, and now the same three buttons that picked 25.5, 51, and 102 seconds select the European scan rates. Note that the European scan rates are indicated in a light blue color below the standard line sequential rates.

When operating in the Volker-Wrasse line sequential modes, you may notice a red line at the left edge of the image. This is because the Volker-Wrasse formats transmit bright saturated red information on the red scan only, which is used to lock the receiving system to the correct sequence of R, G and B. The AVT will also send this red information to help the receiving station, but only in the Volker-Wrasse modes.

For these reasons, Volker-Wrasse modes are more dependable than the US versions of these modes, and are preferred over them when possible.

The "Martin" And "Scotty" Modes

Brief History

Some time after the Robot 1200c scan converter was introduced in Europe, two hams in Europe decided that the eight modes that are normally available in the 1200c could be improved upon.

Accordingly, they independently disassembled the contents of the EPROM that runs the Intel microprocessor in the 1200c and went about doing so. They added various live (in other words, during receive) image processing techniques, and completely threw out the composite methodology that the 1200c normally used. One ham also added a mouse driven menu that the operator could use to select all the 1200c's functions from, including his new ones, and an interface for the mouse itself. Both of these individuals sold these new augmented systems, in the form of new EPROMS, to amateurs everywhere.

Amateurs the world over are grateful to these hams for providing a means by which the 1200c could transfer images through poor band conditions which the composite modes completely failed in.

Perhaps those at Robot Research, Inc. were pleased, perhaps not; these new modes, since they required the 1200c to operate them certainly didn't appear to hurt Robot Research in any substantial way. However, disassembling someone elses copyrighted work and using it as a basis for your own is a criminal offense in this country; These issues are unclear in this case, and perhaps best left undisturbed. This seems to be the inclination of those at Robot, at least to date.

Operation

The AVT system offers two "Scotty" modes, accessible by pressing the button with the scotty dog on it. These modes are known as **Scotty S1** and **Scotty S2** modes. Positive indication that you are in one of the **Scotty** modes is given by the illumination of all three of the R, G and B buttons, and the composite button. Two "Martin" modes are also available, accessible by pressing the button with the Royal Soldier on it. These modes are known as **Martins New Mode M1** and **Martins New Mode M2**. Positive indication that you are in one of Martins modes is given by the illumination of all three of the R, G and B indicators.

To receive in any of these modes, press the proper mode button (**S1/S2** or **M1/M2**) and then the **RX** button, and reception will begin.

To transmit in either mode, press the appropriate mode button, **S1/S2** or **M1/M2**, and then the **TX** button.

All of the Scotty and Martin modes are in the **high** format- 256x240.

Pseudo-Visitel Mode

Brief History

Several years or so ago, Mitsubishi, the Japanese electronics giant, developed a "picturephone" type device. This used techniques that are similar to SSTV techniques to transceive images over the telephone. Unfortunately, they are different *enough* from the AFSK/FM techniques we use to be completely incompatible with the AVT system.

The main advantage of the Visitel mode is simply that it sends a reasonable black and white image over the telephone in just over five seconds. The AVT system can also do this; the same resolution, and the same frame time. The differences are (1) it is not compatible with the actual Visitel unit, and (2) the image quality is much higher than the Visitel's due to the AVT hardware itself.

The specific reason that the actual Visitel mode was not implemented is that Mitsubishi is encoding a great deal of the image information as amplitude modulation, which the limiter circuitry in the AVT front end removes. Transmission of Visitel mode images is possible with the system, and may be implemented at some future date, if there is a demand for such a thing.

Operation

To operate in the AVT's pseudo Visitel mode, select one of the red, green or blue buffers by pressing the appropriate button. You'll find these located next to the CMP (Composite) button you use for the Robot composite modes.

Then press the Visi-Tel button, located at the extreme bottom left of the control panel, and press either the **RX** or the **TX** button, as appropriate. The system will commence the requested operation immediately.

This mode operates in the low format (128x120). Since the images are only 96 by 96, the AVT system expands them to 128 by 120 after reception. Similarly, on transmit, a 128 by 120 image is appropriately modified and then sent out. Some image degradation does result from this procedure.

120 LPM FAX

Brief History

FAX at this line rate is commonly called "Weatherfax" because it is typically used by the weatherfax stations to send maps and satellite photographs. You can always identify 120 LPM FAX simply by the repetition rate - that is, two lines per second, 1/2 second per line.

This is a black and white mode, but differs from most SSTV in a number of ways. Most importantly, FAX is *synchronous*. This simply means that the transmitting station and the receiving station are locked together by the synchronization of very accurate timebases in their equipment. For this reason, should part of an image be damaged during transmission, the rest is not lost or damaged as a result. Once reception of a FAX image has been successfully initiated, it cannot be disturbed by atmospheric or man made interference to the signal itself.

FAX at these rates has been in use for many, many years, and can be found all over the shortwave spectrum.

Operation

First, press the **120** button near the lower right corner of the control panel. You should see the color screen go away, and the workbench screen appear for a moment if you were in an SSTV mode previously.

Once the new, high resolution screen is open underneath the control panel, you may begin reception at any time by simply pressing the **RX** button.

At this point, the control panel is hidden, and the pointer becomes "BUSY". Observe the pointer closely, now. You will see, in the area underneath the word BUSY, one small red dot at the center of the container. This indicates that the AVT is waiting for the *start* signal from the FAX station. Once that has been received, the single dot will change into two dots in the edges of the container, indicating the AVT is now waiting for the *phasing* signal from the FAX station. Once the image has been phased, reception will begin, and the dual red indicators will extinguish.

If the AVT fails to start receiving due to mistuning or poor signal conditions, you can press the left mouse button once, quickly. It will proceed to the phasing state. Similarly, if the AVT fails to phase the image (which also prevents reception), you can bypass the phasing state by one more quick press on the mouse button.

If the image is started automatically by the system, then all you have to do is wait, and after 10 minutes, the system will return to its "wait-for-start" condition. At this time, you may press and hold the left mouse button for about 10 seconds, and the system will return the control panel to view.

If you have enough memory (over 1.5 megabytes), the AVT will have received an image at a resolution of 1024 by 1200 lines. This is very high resolution, in fact more than the screen can display to you at one time. To allow you to see all of the image, you can pan around it by simply pressing the cursor keys. You will also have access to the **FAX Control Window**, which makes a number of useful editing commands available to you.

If you have a one megabyte machine, then the image was received in 640 by 400, and what you see is what you get. All portions of the image are visible on the screen at one time. The **FAX Control Window** will not be available, since there is not enough memory for it to perform any operations upon the received FAX images.

60 LPM FAX

Brief History

This FAX line rate is most often used by news services, and in fact is often called "Newsfax". The line rate is 60 per minute, or one per second, easily distinguished from the 120 lpm of a weatherfax signal.

For reasons which are unknown, newsfax stations rarely, if ever, send a start signal. They begin their transmissions with a phasing pulse, and then launch right into the image.

Other than that one format difference, and the rate at which the lines are transmitted, newsfax is extremely similar to weatherfax.

At one time, there were a large number of newsfax stations on the air; but at the time of this writing, we know of only one, transmitting at 20.737.88 MHz, lower sideband. See **Appendix C** for other frequencies.

Other stations besides news services do use the 60 lpm rates; the Japanese and the Russians both operate what we would call weatherfax at 60 lpm at various locations within the shortwave spectrum.

Operation

To operate this mode, you should follow exactly the same procedures as those for 120 lpm FAX, *unless* you are receiving an actual newsfax station, such as the one on 20.737.88 MHz. In this case, you should open the **SET** window by pressing the appropriate button, and enable the **Phase** button (indicator red) and then close the **SET** window. This tells the AVT system to *not* look for a start pulse, but to instead begin reception upon receipt of the phasing sequence. This function also affects the other line rates, so be careful you don't forget that you have set this mode.

240 LPM FAX

Brief History

This FAX line rate is used almost exclusively on satellite transmissions, and not within the usual shortwave bands. These transmissions are not usually FM FAX at all, but instead are AM FAX. The 240 LPM scan rate effectively reduces the resolution to 800 pixels per line, maximum - which is a shame, considering that images of the earth from above hold much interest for most of us. We call this FAX mode GeoFAX.

You can find GeoFAX images in several places. First, on 137 MHz in the VHF band, coming from various US and Russian polar orbiting satellites. Next, you can find them up in the GHz bands coming from geostationary satellites - these stay over one place all the time, and so comparing images from them is very interesting. Lastly, you can find them in the video output passband of your satellite television receiver.... surprising, but true. To receive these, you simply attach a SW radio to your TVRO system at the video output, and tune around. It depends on which satellite, and so on, but there are certainly many interesting things "lurking" in there. See **Appendix C** for details on signals that are currently available on TVRO systems.

Operation

Since the AVT system is an FM based system, it is not capable of receiving these images on a stand-alone basis. However, there is relief for the terminally curious.

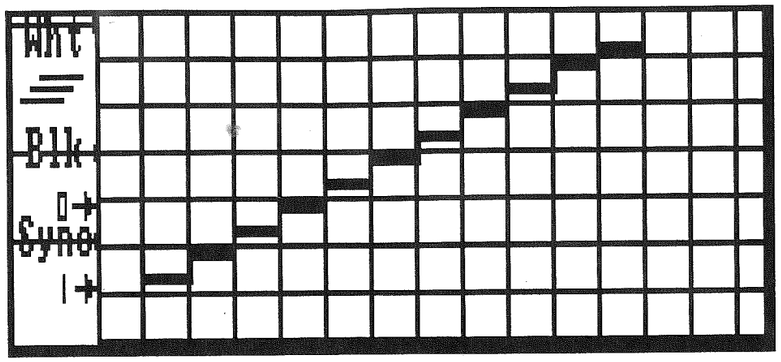
Refer to **Appendix G** for the address of Overview Systems, which supplies an accessory board for the AVT system which converts AM FAX to FM FAX, as well as other convenient functions.

User-Defined Demodulation Curves

This function is only active when you are in FAX mode.

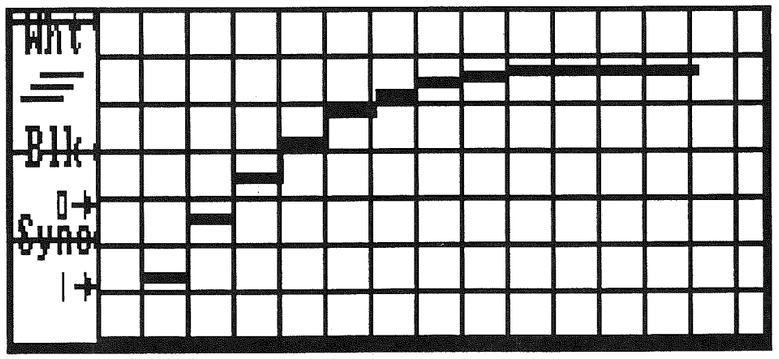
A demodulation curve is a *transfer curve*. It allows you to assign the incoming signal a new value, according to a curve you create. In FAX mode, the AVT "sees" the incoming grey levels as 64 different values. When these are used to create a FAX image, we need only 16 values.

A linear demodulation curve causes every four incoming values to one output value. Examine the following graph.



Graph of linear transfer curve

This is how the system initializes. You can, however, change the curve to suit differing needs. For instance, on a satellite image, a great deal of information is contained in the lowest grey levels. You can change the demodulation curve to enhance these grey levels by mapping them to brighter components of the output side. To see how such a demodulation curve might look, examine the following illustration.



Graph of Log demod curve

You can create these curves very easily. Just select **Generate Curve** from the FAX menu, and move the mouse over the scope in the shape you want the curve to take. When you are happy with the display, click the left mouse button.

Once a curve is defined, you can save it using the **Save Curve** selection from the FAX menu. They can be reloaded any time you are in FAX mode by selecting **Load Curve** from the FAX menu.

Several curve files are supplied with the system so that you can see what types of curves are useful.

Curve files themselves are actually text files which can be edited. These files are very simple in format - when read, the AVT reads 64 different values from the file, one for each received grey level. You can place values on the same line, or on different lines. Again, examine the supplied files for examples.

Grabbing Images From Other Programs In The Amiga

This function is available in both the SSTV and the FAX menus. It is the heart of image transmission from the AVT system.

The idea here is that the AVT system can retrieve an image from any other screen that is currently active, or open, in the Amiga when GrabScreen is selected. This includes images shown with public domain programs like BackShow, SuperView or Ushow, and images from games, the WorkBench, or any other program source you can think of. The only limitation to this procedure is the amount of memory you have in your Amiga.

You also need to keep in mind that the Amiga has two types of memory - **chip** and **fast**. Chip memory is where all screens are built, and it is a limited resource. At this time, most Amiga systems have a maximum of 512k of chip RAM. Newer machines are coming from the factory with the capability of one megabyte of chip RAM, a welcome relief from this bottleneck. The fast RAM resources in your Amiga are almost unlimited by comparison - the lowliest model in the Amiga line can have up to nine megabytes of fast RAM, and so if it's a problem, you can always get more.

NOTE: At the time of this writing you can upgrade your Amiga 500, 2000, or 2500 to one megabyte of chip RAM by having the "Fat Agnus" chip installed by your Amiga dealer or service center. The chip costs about $120.00 and the installation costs may be in the range of about $30.00; we strongly recommend that you have this done if possible, as it will make the operation of the AVT system much smoother. In addition, there is at present a project under way for Amiga 1000's to be able to use the Fat Agnus chip and so have one megabyte of chip memory like the more recent models in the Amiga line. For details on the a1000 upgrade, contact Greg Tibbs on the Compuserve network in the AmigaTech forum.

Something to keep in mind is that when you run out of fast RAM, the Amiga puts everything into chip RAM - this is undesirable because programs that are in chip RAM will run quite slowly under many common circumstances. When in doubt, get more fast RAM; it's a purchase you will not regret.

Make **sure** that you have copied the BackShow and NextScreen programs from the Tools drawer on the AVT disk into your "C" directory on your WorkBench disk or your hard drive before you try this example.

This is the procedure to follow to grab Amiga screens:

1 - Make certain you have a CLI window open.

2 - Select "Grab Screen" or "Framed Grab" from the menu using the right mouse button and standard menu operations.

3 - Enter "BackShow PictureName" in a CLI or click image ICON if the image's ICON is a "Project" type ICON and it's "Default Tool" has been set to "BackShow", and wait for the image to be displayed. You must NOT try to BackShow the image using either of these methods BEFORE you perform step 2, above, or you will most likely run out of **chip** memory.

4 - Using **left-amiga-M**, bring the control panel to the front; This capability depends upon your having properly installed the utility **NextScreen** (supplied in the **For C** directory). If you have not installed NextScreen, then you will have to use the screen's front and back gadgets, located at the upper right hand corner of each of the Amiga's screens.

5 - Press the button labeled **Grab It** and wait....

6 - When requested to, **close** the BackShowed image by clicking at the upper left of the displayed image - you may have to use left-Amiga-m to get back to the image's screen. If you do not close the image before continuing, you will probably run out of **chip** memory and have to re-start the program.

7 - Then press the "Ready To Continue" button and save the image to disk using the **Save SSTV** menu item.

The AVT system can grab video from any Amiga screen - at least, all the standard screens. This includes:

320x200	2 color
320x200	4 color
320x200	8 color
320x200	16 color
320x200	32 color
320x400	2 color
320x400	4 color
320x400	8 color
320x400	16 color
320x400	32 color
640x200	2 color
640x200	4 color
640x200	8 color
640x200	16 color
640x400	2 color
640x400	4 color
640x400	8 color
640x400	16 color
320x200	Hold-and-Modify (HAM)
320x400	Hold-and-Modify

This means that you can grab your WorkBench screen and send that image right on out... You can even send the control panel of the AVT system, if you like - it *is* on its own screen.

You'll notice that there are two grab entries in the project menu. One is **Grab Screen** and one is **Framed Grab**.

Grab Screen attempts to take the **entire** image on the screen being grabbed and make it fit in the resolution you were in when you selected **Grab Screen**.

Framed Grab takes the source screen pixel-by-pixel and captures it into the resolution you are using (128x128 or 256x256) regardless of the resolution of the source screen. **Framed Grab**, when attempting to grab image information from a screen, offers you the opportunity to control the horizontal position that the grab begins at. For instance, in low-res modes, you are going to get 128 pixels, total. This option allows you to specify where on the screen these 128 pixels are taken from. Choices include flush left (0), 128, 256, 384 and 512 pixel positions. Currently, vertical positioning is not supported, but it is planned for the future.

If you use the GrabScreen program supplied instead of the menu function, simply follow the prompts. You can grab in high or low resolution, from HAM or normal screens, and if a HAM screen, the option to grab pixel for pixel is also offered there.

You can run the GrabScreen program from either the CLI or the WorkBench.

Let's say you are in high resolution (256x240). If you do a regular grab of a HAM screen, the program takes the 320 by 200 and compresses it into the 256x240. As you might well imagine, some information is lost; there just aren't enough pixels, even in standard hi-res SSTV, to represent all the data on a screen as dense as a standard HAM screen.

In Framed Grab, the program will take the *first* 256 pixels from the left side of this HAM screen, and map those directly into the SSTV image. This means that image information on the right is lost; but the information that WAS taken is 100% accurate - pixel-for-pixel. If you use the AVT modes, you'll find that you can send sections of your HAM screens all the way around the world, and the recipient will be able to read the text you might have typed in there - in any font at all.

Of course, you can use the 94-second mode - that is 320 pixels wide, and you will get all the pixels.

Image Processing And Modification

The AVT system offers a large number of different options for changing images to suit your preferences. These include items that range from sophisticated image processing to drawing mustaches on your friends faces. This section of the manual goes through each of the various options, one by one.

Tinting

This operation is mainly used in conjunction with the Robot composite SSTV modes, although it can be used with any other mode as well, after the proper setup.

The tinting control is in the **SYNTHESIS** window, and consists of a single proportional gadget (slider). This gadget is labeled at the ends **RED** and **GREEN**. Accordingly, if the slider is moved towards either end, the image will be tinted in that direction.

You should keep in mind that the Amiga is "awake" all the time, and processing all your mouse button clicks and other activity like keystrokes. Since the tint operation can take several seconds to complete, *don't* click again on the control until it has "snapped" back to the center, indicating that the current operation has completed.

For actual use, if an image is too red, you want to tint it green, and vice-versa.

You may only use the tint control under two conditions; however, using it indiscriminately may destroy your image completely. Here are the two allowable situations:

1: Just after a Robot mode image has been received, the composite modes *only*; 12-, 24-, 36- and 72-second color.

2: Just after an image has been prepared for transmit, using one of the four Robot composite mode synthesis buttons in the **SYNTHESIZE** window.

It is in the use of the second item above that allows you to tint virtually any image. The procedure is to prepare the image for transmit using either the **24** (for low format images) or the 72 (for high format images) - even though you are **not** planning to transmit in these modes. Once they have been prepared for transmit, they can be tinted, and once tinted, sent in any mode you care to use, including AVT modes. One thing to be careful of here is that the format of the source image *must* be either low or high. Tinting is not effective on AVT, AVT high, or AVT super format images.

Brightness

The AVT offers a means to adjust the overall brightness of an image that works similar to the brightness control on a monitor or television, except that the image itself is modified. The brightness control works on color or black and white images. When you use this feature, ensure that you have the correct memory selected (R, G, B or Composite) *before* you attempt to change the brightness, since the system will affect only the memory that is selected.

To operate the brightness control, open the **Synthesis** window by clicking on the Synthesis gadget. Observe the lower right hand corner of the window, and you will see two arrows with the letter **B** in them. These are the brightness controls. One arrow points up - this is the arrow that *increases* the brightness of an image. The other arrow points down, and will decrease the brightness of an image when pressed.

It is a good policy to use the *Histogram* function before making any changes to the brightness of an image. This will provide you with specific information about the image that is useful in making decisions about just how bright (or dark) you wish to make the image. See the section on Histograms for more information on their use.

Contrast

The AVT offers a means to adjust the overall contrast of an image that works similar to the contrast control on a monitor or television, except that the image itself is modified. The contrast control works on color or black and white images. When you use this feature, ensure that you have the correct memory selected (R, G, B or Composite) *before* you attempt to change the contrast, since the system will affect only the memory that is selected.

To operate the contrast control, open the **Synthesis** window by clicking on the Synthesis gadget. Observe the lower right hand corner of the window, and you will see two arrows with the letter **C** in them. These are the contrast controls. One arrow points up - this is the arrow that *increases* the contrast of an image. The other arrow points down, and will decrease the contrast of an image when pressed.

It is a good policy to use the *Histogram* function before making any changes to the contrast of an image. This will provide you with specific information about the image that is useful in making decisions about just how much of a contrast change you wish to make to the image. See the section on Histograms for more information on their use.

Histograms

The histogram is a tool that has been in use for many years in the field of *Image Processing*. It can provide an extremely powerful and useful set of information to the user when its function is fully understood.

To create a histogram of an image, first ensure that you have the correct memory selected - red, green, blue or composite. This tells the system what type of image you want to have the histogram taken from. Then press the **Synthesis** button to open the synthesis window. You'll see a button at the lower left corner of the synthesis window marked with an **H**. Press this, and the histogram will appear in the scope area.

Now that you have the graph displayed, you need to interpret it in such a way so that it aids you in processing or changing your image. There is no easy way to explain how to do this, but experience will help. We recommend that you run the histogram on every image you have a chance to.

After a while, you'll notice that images that are too dark tend to have graphs that "cluster" towards the left hand side of the graph. Images that are too bright cluster towards the right hand side of the graph. Images with low contrast don't fill the graph, and images with too high a contrast will extend off the edges of the graph. When you run the brightness or contrast operations, take *another* histogram after the operation - this will show you exactly what these operations do to the images.

How the Histogram works

When you have an image in the AVT System's memory, the computer sees the image a group of pixels, while *you* see it as an "image". In the AVT system, a pixel may be one of 64 possible grey levels - in a color image, there are three memories, the red, green*B,* and blue memories. Here, it is not levels of grey that we are really discussing, it is levels of red, green, and blue. Otherwise, the concept is the same.

When you ask the system to create a histogram of a black and white image, first it creates 64 individual empty variables which you can think of as buckets. These buckets are all set to zero to start with, and then the system examines the picture in memory. Each pixel is examined individually. When the pixel is examined, it will be found to have a value between zero and sixty three, no matter what kind of image is in memory since these are all the possible values for a pixel. Let us say that the first pixel examined has a value of 30. The computer takes the value in bucket (variable) number thirty and adds one to it. Then it moves to the next pixel in the image, examines it to see what the brightness of that new pixel is, and adds one to the bucket that matches the brightness value. This process continues until the entire image has been examined in this manner.

When this process is complete, each bucket (variable) represents the number of pixels in the image that have the brightness that is equal to the bucket number. The histogram graph itself is a display of the values contained in all 64 buckets from zero on the left to 63 on the right.

One thing to note here is that an image may contain literally thousands of pixels of one level, for instance black if it has a black background, and only a few of another level. Since we can't very well show you a line thousands high on the screen, the values in the histogram are *normalized* against the bucket with the largest value. This is a very simple process. If the bucket with the largest value has 25,000 in it, then the top of the scale indicates 25,000, and the center of the scale indicates 12,500 and so on. This process is fully automatic.

In the case of a color image, there are three separate histograms taken - one for each of the red, green and blue buffers. You will notice that once each of the three graphs are generated, they are redrawn. This is the same normalization process, however all three buffers are checked against each other for the maximum color value and then are normalized accordingly before being redrawn.

Zooming

To zoom an image, open the Synthesis window, and press the button with the Z on it. The HAM screen is brought to the front of the display with a box drawn on it. Move the box around, and when you have it positioned over the area you want zoomed just press the left mouse button. After a short delay, the zoomed image will be drawn on screen.

Make certain that you have the correct mode selected before you do the zoom, as well as the correct selection of R, G, B or Cmp buttons so that the zoom operation affects the correct memory(s) and works in the correct resolution.

Luma Generation From Color

This operation takes a color image and changes it to a black and white image using one of the two following formulas:

Pixel = (.11 x Red) + (.30 x Blue) + (.59 x Green)

Pixel = (Red+Green+Blue)/3

The first formula creates an image that should be the same brightness as a black and white image on your television. This combination of the various red, green and blue components causes your eye to respond in the same manner, brightness-wise, that it would if the image was still in color, or so scientists tell us.

You might use this operation to take an image that was sent to you in color, and modify it so that you could send it on to a station that has only black and white capability. One interesting side-effect of this operation is that noise in the image is somewhat reduced by the luma conversion which acts like an averaging operation to some degree.

The second formula is useful when a black and white image has been sent using one of the AVT modes. It averages the three buffers together, resulting in a marked reduction of apparent noise spots.

Mode Conversions

A mode conversion takes an image from one resolution and changes it to another. When going from 128 to 256, the system creates "new" pixels that are generated by taking the average of the pixels surrounding them in all directions. When going from 256 to 128, the system takes every other pixel from every other line.

In addition to the standard resolution conversions, you can convert 200 line AVT images (94 AVT) to 256 format images, and then on down to 128 if you need to. This is done by pressing the "94" button in the **Synthesize** window, and optionally following that with a press of the **128** button.

Line Corrections

Line correction is a very powerful means for repairing damaged SSTV images. The line correction facilities in the AVT System allow you to reposition individual scan lines horizontally, replace them from the line above, below, or an average of both, adjust the red, green or blue buffers up or down by a scan line, and shift the buffers around between themselves starting at a scan line of your choice.

To access the line correction functions, press the Synthesis button, and when the Synthesis window opens, press the **Lcor** button at the upper right corner of the Synthesis window.

A cursor appears beside the image to be corrected. This cursor is a very small white dot. You may use the arrow keys on the keyboard or the up and down arrows in the window (with the mouse) to position the cursor beside the line you wish to modify. The sideways arrows (keyboard and window) allow you to move the line horizontally. If you are not certain if you are on the correct line, the easiest way to check is to move the line one position left, and then right again. If the line you want moves out and back again, you're positioned correctly. If not, move the cursor accordingly. Notice that the simulated LED display indicates the line number - if you'll be working on this image again, you may wish to take note of this, as it makes it very easy to return to the same line again later.

Once you are on the correct line, you can press any of the correction function buttons to perform an operation that affects the image at that line, or beginning at that line.

The three red, green and blue arrow pairs are controls that move the contents of that specific buffer up and down by one scan line each time the arrow is pressed. Note that the buffer is only moved starting at the line the cursor is on and the rest of the way down the image. This allows you to repair images that have single misaligned frames. This type of problem often occurs in frame sequential types of images.

The button marked **Shft**, which is short for "Shift", takes the contents of the R, G and B buffers and literally shifts them among themselves starting at the current cursor position and continuing on downwards thru the buffers. The red scan lines are moved into the green buffer, the green into the blue, and the blue into the red. This capability is provided to enable you to repair "out-of-color-sync" conditions that can occur when operating the Volker-Wrasse line sequential modes or the WA7WOD line sequential modes. There is no other application for this function that we are aware of.

The three buttons with graphic symbols on them are the line replacement controls. The two buttons with the letter **C** and an arrow on them are line **copy** controls. These controls copy the line above (or below) to the line that has the cursor located on it. The third control allows you to replace the line that the cursor is located on with one that is the average of the line above and the line below. This button has two arrows pointing to a line in the center of the button. Try each of them to see what differences you get.

The **Range** button allows you to run the copy and average operations over a range of lines. To use it, place the cursor on or above the line where you wish to begin the replacement. Next, press the Range button. Now move the cursor to the last line or just past the last line where you wish the replacement to occur and then press one of the three controls (copy up, copy down, or average). The system will replace every other line in the image using the method you specified with the control buttons. This operation can be used to do an excellent job of repairing SSTV images that have been damaged when sent in the AVT QRM modes.

Since the AVT QRM modes send all the even lines first, and then the odd lines, it is evident that no line that is next to another line on the image is actually *sent* next to that line. In fact, if you think about it, adjacent lines are all sent exactly one half of the frame time later. This means that if there is a burst of noise, it affects *every other line*, rather than all the lines in one area of the image. Because of this, once the image is in memory, you can re-create the missing lines using the Range and Average buttons in one easy step. The resulting image is generally close to the original, often so good that the sending station won't notice the change!

With a little bit of practice, this operation will become second nature to you.

Frame Corrections

There are many situations where an image is damaged overall; by this we mean that it's quality is low, perhaps due to a high noise level on the radio or telephone, or "splatter" from nearby stations on the radio.

When this occurs, the system offers you the capability of processing the entire image in such a manner as to remove spot or burst noise.

To access the frame correction functions, press the main panel button that has the dustpan and the whiskbroom drawn on it. This icon is intended to convey the concept of cleaning up an image.

In the frame Correction window there are three basic types of functions. The first is one that provides a function that can clean up most images amazingly well. It is controlled by four of the buttons in this window. One button is a "knob" in a proportional gadget which adjusts the severity or intensity of the cleanup operation from 0 to 15, where 0 is the least intense, and 15 is maximum. The other three buttons are marked with geometric shapes that resemble an **X**, a **+**, and a **snowflake**. These three buttons cause the system to execute a procedure that examines every pixel in an image and then possibly replace it with the average of some of the surrounding pixels.

The replacement operation proceeds as follows. First, using the pattern that is marked on the button you chose, four or eight pixels are added together and then this sum is divided to create an average value that represents the general intensity of the image in the area of the pixel in the middle of this geometric shape.

Next, the pixel in the center is examined to see how much it differs from the generated average. For instance, if the average indicates that the intensity expected here is **eight**, and the pixel in the center has a value of **ten**, the difference is **three**. With the intensity set to maximum, this pixel will be replaced, because the process has been told that very little difference between the calculated average and the actual value is to be accepted.

If you set the intensity to minimum (0), then pixels will be allowed to diverge by as much as 16 (out of 64) grey levels before they will be replaced.

As a general rule of thumb, you can use a setting of zero, and the **X** shaped button to do your processing. This modifies correct image information the least, while catching obvious noise spots and repairing them.

QRM *Incorrect To Correct* Mode Corrections

These operations are designed to address two possible types of errors in the receipt of AVT mode images. To explain the situation, a brief review of the line ordering in an AVT mode image is required.

Standard AVT images send the lines in an image sequentially, from the first to the last in the order you would expect. These images are received and displayed without any extra processing on the part of the system.

QRM mode AVT images are quite different from this, however. QRM mode images have all the even lines sent first, and then all the odd lines. Once the lines have been received, the system places them in their correct position in the image, which involves re-sorting the odd lines in between the even lines. There are extremely good reasons to do this, but they are not important for the purposes of this discussion.

The two problems that can occur involves the line ordering.

The first would occur if, for instance, you are in QRM mode, and you *manually* start reception of an AVT mode image. The image looks like it comes in full screen, but after the reception is complete, the image is re-drawn such that it looks quite scrambled. This is because the system has taken all the lines in the second half of the image and placed them *between* all the lines in the first half of the image, on the assumption that the sending station was in QRM mode, when in fact they were not.

To fix this type of problem, press the gadget in the Frame Correction window that has alternating blue and yellow lines.

The second occurs in the exact opposite situation. You would *not* be in QRM mode, but the sending station is, and what you see during reception is two short images (the even lines, followed by the odd lines), and when the system displays the image, you *get* two short color images. Here, the sending station has sent the lines even first, then odd, as required by the QRM mode, but you were not *in* QRM mode, and so the problem occurs.

To fix this type of problem, press the gadget in the Frame Correction window that is half yellow on top, and half blue on the bottom.

The two gadgets discussed here actually look like the image does, and so are very easy to remember. If the screen looks like lines that are alternating every line, press the gadget that alternates every line. If the image looks like two short images on top of each other, then press the gadget that looks like two short images on top of each other.

QRM Offset Corrections

One other type of problem that can occur when using AVT QRM mode is an *offset* problem. It will only occur in QRM mode, and only when you, or the other party, has an AVT unit (the actual hardware) that has drifted off frequency.

The visual result is that the image looks like every other line is "shifted" sideways by one or more pixels. To correct this, use one of the two **arrows** that are marked **QRM Offset** in the Frame Correction window.

The reason that this problem occurs is obvious after a short explanation. Since the AVT modes are completely synchronous, that is, they do not use a **sync** pulse on each line to re-register the horizontal position of each line, they AVT hardware must provide *very* precise signals to the Amiga so that the Amiga knows when to begin receiving on the next line.

If the AVT hardware is just the slightest bit off in its timing, it is possible that when an image is partway down the screen during reception that the scan lines will begin to drift to one side or the other. Now, you're generally not likely to notice a drift like this if it's only one pixel-after all, that's only about 1/3 of one percent over the entire image.

However, if you are operating in QRM mode, consider the following. Since QRM mode sends all the even lines first, which amount to one half the image, and then the odd lines which are the other half of the image, an offset causes a problem.

When the image is reconstructed by placing the odd lines in between the even lines, the odd lines (which come later in the transmission) will have that one pixel displacement. If you think about it, the first even line is sent first, and the first odd line is sent first *after* all the even lines, making a difference in time of exactly one half the frametime between them. The second even line and the second odd line will have *exactly* the same difference in time, 1/2 the frame time between them, as they come after the other two. If the drift of the AVT hardware causes a one pixel difference after one half the frame, then all the odd lines will be offset by one pixel sideways.

The reason this is easy to fix is that all the odd lines will be offset by the same amount, so moving all the odd lines sideways by the same small amount will completely correct the image. That is what these two arrow buttons do for you.

We hope that you will see few images like this, as all units are correctly adjusted when they leave the factory, but over time they will drift. If this occurs, or you suspect that it has occurred in your unit, refer to **Appendix H** at the end of this manual for the alignment procedure.

Color Bars And Grey Scales

The AVT System offers two kind of standard reference bars that you may place either under or over any image. One bar is a color bar, containing a range of colors. The other is a grey scale, containing a range of greys.

To place these bars onto an image, you use the blue arrow in the main control panel to specify the top or bottom as the location you want the bar to be placed into. If the blue arrow points up, then the bar goes on the top of the image. If the blue arrow points down, then the bar will be placed at the bottom of the image. The bars themselves are placed by pressing either the color bar or the grey scale gadgets in the main control panel. You will find them right below the arrow gadget.

It is also possible to generate a full-frame color bar and grey scale combination image by pressing the larger square gadget just to the left of the **synthesize** button.

As with all image processes, ensure that the AVT system is in the correct mode *before* you place color bars or grey scales so that you do not unintentionally damage your image.

You can create custom color bars for use in the SSTV modes by using any AMiga paint package. The first 256 pixels out of each of the top sixteen lines in the picture you draw in the paint program will be turned into a custom color bar if you follow these steps:

1) Draw the color bar (top 16 lines, left 256 pixels) utilizin your paint program, and save the image as an IFF file. terminate the paint program.

2) Execute the AVT software from its icon normally.

3) Select the **90 AVT** mode.

4) Select **Framed Grab** from the SSTV menu.

5) Follow the requesters... use Backshow to display the IFF image you saved that contains your custom color bar drawing.

7) Select **Create** from the Color Bar submenu in the SSTV menu.

8) (optional) Save the color using **SAVE** in the Color Bar submenu.

Painting

The AVT System offers a minimal paint mode, intended for scribbling mustaches on your friends faces and other quick operations. It is **not** intended to be a full featured paint program - for this capability, we highly recommend one of the following commercial products:

DPaint III or DPaint II for normal color images
Photon Paint 2.0 for HAM mode images
DigiPaint 3.0 for HAM mode images

NOTE: DPaint II and DPaint III are trademarks of Electronic Arts corp, PhotonPaint 2.0 is a trademark of MicroIllusions Inc, DigiPaint 3.0 is a trademark of NewTek Inc.

To operate the paint capability, select the color you wish to use by sliding the three colored proportional gadgets below the scope to a position that produces the color you want. Then press the button just to the right of these sliders that has the paint brush on it, and draw on the SSTV image. When you are satisfied with the results (or are done, in any case), press the paint brush icon again so that the red indicator in its upper left hand corner goes out. If you do not do this, some operations will not function correctly until you remember and turn the paint mode off.

Adding Text

The AVT provides a comprehensive font handling system that is compatible with the Amiga's standard font system. It allows you to use fonts as tall as 50 points on any image, with text preview, italics, bold, underline, coloring and shading.

There are three windows that can open when you are dealing with the text overlay capabilities in the AVT system. The first window that opens is the font selector. When you begin operation of the AVT system, no font is selected, and so the font requester is presented to you automatically the first time you try and write text on anything. Once the program has read the fonts in your system, you can scroll through the list and choose the font you want to use.

If you wish to look at an entirely different bunch of fonts, just select the **New Fonts** item from the Text Control menu. This selection flushes all fonts in memory, and reopens the font selector window again after re-reading whatever fonts are found.

The Amiga always keeps its fonts in a directory that has a logical assignment of **FONTS:** and this is where you will find the fonts on your WorkBench disk. One interesting feature of the Amiga is that if you insert a disk that has the same name as a logical assignment, in this case a disk named **FONTS**, the system will no longer *use* the logical assignment - instead, it looks at the disk.

This provides a very handy means to change fonts by simply inserting a disk in your computer. If you wish to use the fonts on your workbench (which we assume is named **WorkBench 1.3** or something similar) then the system automatically looks in the **devs/fonts** directory. If you replace your WorkBench disk with a disk named **FONTS**, however, then it looks on that disk for your fonts. So you can use the WorkBench fonts, or a set that you have placed on another disk. You may also place the disk named **FONTS** in another drive, and leave the WorkBench inserted - the system will still find the fonts on the new disk, and ignore the fonts on the WorkBench. Once the new disk is inserted, remember to choose **New Fonts** from the Text Control menu, or the AVT system will continue to look on the WorkBench disk!

When you have chosen a font, the system will open the text preview window. A cursor appears in a text box, and all you have to do is type your text and press the **return** key. The text immediately appears in the preview area in the font you selected. If you would like to see how this will look with bold, italic, underline or any combination, click on the associated gadget and observe the text, which is redrawn immediately. Notice the **caliper** that appears just over the text preview area. This is a line that is marked with the numbers 128, 256 and 320. These points on the line indicate the maximum width available in the different SSTV modes. If you exceed the widths marked here, the text will not render correctly in the final image. Try it and see!

When you are satisfied with the appearance of the text, press the **DO** gadget, and the third window appears. This window allows you to color your text in a number of ways, and then overlay it on the target image. To render you text in a single solid color, click on the box underneath the word **Solid**, and then adjust the red, green and blue sliders until you are satisfied with the color in the Solid box. Now press the box that is over the word **Sld**, and the control panel will disappear. A box shape will appear that you can move over the target image until the text is placed where you want it, at which time you press the left mouse button. The text is rendered into the memory, and then the image is drawn for you with the text in it. If you do *not* like the results, press the **UNDO** button, and the text will be removed so that you can start over.

You can also generate text using the color sweep functions. There are two sweep operations, one sweeps two chosen colors from top to bottom of the text, while the other sweeps it from left to right across the text. To use the sweep functions, you click on the start box, and set the starting color using the sliders. Next, click on the End box and set its color the same way. Now press one of the two sweep gadgets (at the top left of the text window) to generate the text into your image. As before, if you do not like the results, you can use the **UNDO** button to remove them from your image.

There is one miscellaneous function available in the text window, intended for generating backgrounds for your text screens. This is a memory fill operation, which uses the **solid** color as it's source. You pick the solid box, set the color, and then press the FILL button. This fills the entire image with that one color. It is useful if you would like a background of a particular color to write text upon.

Miscellaneous Options And Features

The Status Window

The item called **Status** in the project window will provide you with important information about the availability of memory in your Amiga while the AVT is running. There are five individual items in the window, total chip memory available, total fast memory available, largest "hunks" of each of these types of memory, and the total of chip and fast memory.

The **Largest Hunk** of **Chip** memory is probably the most important item in the window, because when you display an IFF image using **BackShow** to be "grabbed" it is loaded into **chip** memory by the Amiga. In order to display a particular type of IFF image, various amounts of *continuous* chip memory are required. Amiga images are made up of various numbers of **bitplanes** of differing sizes. When in doubt, refer to the following table.

Mode	colors	Chip Memory Required
320x200	32	40,000 bytes in 5-8,000 byte hunks
320x200	16	32,000 bytes in 4-8,000 byte hunks
320x200	8	24,000 bytes in 3-8,000 byte hunks
320x200	4	16,000 bytes in 2-8,000 byte hunks
320x200	2	8,000 bytes in 1-8,000 byte hunk
320x400	16	64,000 bytes in 4-16,000 byte hunks
320x400	8	48,000 bytes in 3-16,000 byte hunks
320x400	4	32,000 bytes in 2-16,000 byte hunks
320x400	2	16,000 bytes in 1-16,000 byte hunk
640x400	16	128,000 bytes in 4-32,000 byte hunks
640x400	8	96,000 bytes in 3-32,000 byte hunks
640x400	4	64,000 bytes in 2-32,000 byte hunks
640x400	2	32,000 bytes in 1-32,000 byte hunk
320x200	4,096 HAM	48,000 bytes in 6-8,000 byte hunks
320x400	4,096 HAM	96,000 bytes in 6-16,000 byte hunks

"Big" Picture

This option is set either from the CLI's command line, WorkBench ToolTypes, or a button in the **SET** window. It changes the way the system displays low format images from one fourth screen to full screen. The only disadvantage is that it takes a little longer to display a full screen than it does a one fourth screen. There are two competing factors involved in the size of low resolution image.

First, in favor of a one fourth screen display, is that a low format image has (generally) about 128 by 120 pixels, which fits quite well into one fourth of a 320 by 200 Amiga HAM screen.

However, in favor of a full screen image, is that HAM mode will sometimes take several pixels to change from one color to another, and when you display a low res image full-screen, each pixel from the image gets multiple pixels on the screen, and so change colors more effectively.

Try it both ways, and use the method that suits you best.

Compressing SSTV and FAX Format Files

Normally, the system saves SSTV format files as three bytes per pixel. Of these three bytes, six bits are used in each- that creates files that have 256,000 colors stored in them.

When you turn the **compress** option on (from the CLI as a command parameter, the WorkBench icon as a ToolTypes entry, or the button in the **SET** window during AVT operation) SSTV format files are saved as three "nybbles" instead of three bytes. In addition to this, some compression of the data is done. Images saved in this manner will not degrade at all from your perspective as an Amiga user, as this still saves 4,096 colors, all a stock Amiga can display. Total disk space used will generally be well under one half the space that a non-compressed file takes.

Individuals with other types of display capabilities, like the Robot 1200c scan converter which has a 256,000 color display, may be able to tell the difference, although Amiga-to-Amiga transfers will work perfectly.

Keep in mind that Amiga images, those grabbed from Amiga screens using the GrabScreen capability, have only 4,096 colors to begin with, so there is no particular advantage to saving them with the 256,000 color file format. In short, we suggest that you keep the **compress** option on, for the most part.

FAX images are normally saved as 4 full bitplanes of 153,600 bytes each, for a total storage requirement of 614,400 bytes. When the **compress** mode is on, fax images are saved with special run length compression techniques. This can take longer when saving the image, but generally brings about a decrease of total load time when the file is reloaded. No data is lost due to the compression scheme utilized.

Macro And Script Operation Using ARexx(tm)

Note: ARexx is a trademark of Bill Hawes

The AVT system is unique in many ways, but perhaps one of its most sophisticated and useful features is the ability to create short macros and arbitrarily long, complex external scripts that manipulate the program's facilities in preset ways.

ARexx is short for Amiga Rexx, Rexx being a computer language that was developed for interprocess communication on IBM mainframe systems.

ARexx has a number of fascinating capabilities, among which are:

1 - **Interprocess communication:** This allows two programs that both have ARexx capabilities to talk to each other. For instance, Your AVT system can send messages to and from your database system, your spreadsheet, your radio control program and your packet program.

2 - **Macro capability:** This provides a means where you can set up a short sequence of steps to perform a commonly used operation or operations "in" a function key.

3 - **Script capability:** This lets you set up an external text file, using a text editor, that can have any number of complex steps in it, the only limitation being the size of your computer's memory. The name of the script (instead of the commands themselves) is placed in the macro key.

4 - **It's a Complete Language:** ARexx has everything from floating point math, date and time capability, execute capability, loops, if's, do's, while's, subroutines, calling of other scripts, multiple scripts at one time (multitasking), recursive capabilities... it's more powerful than BASIC in many ways.

ARexx, as implemented in the AVT system, allows you to run either short macro commands, or long, complex scripts. All the other features of ARexx are available as well, of course.

For details on the ARexx language itself, refer to the ARexx manual. Much of the following material in this section will not make much sense if you are not able to refer to the ARexx manual, or do not have a working knowledge of the language already.

The AVT allows you to use either the Amiga's menu system or the keyboard to manage all the ARexx facilities.

From the **MACRO** menu, you can execute any one of 10 available scripts or macros; define or re-define them; save and load them in sets of ten; invoke your choice of screen editor (ed, UEdit, CED, uEmacs, etc) on any script in the set of 10; define or re-define the editor to be used; and execute any ARexx macro or script directly.

From the keyboard, you can perform these same operations. To execute one of the 10 available ARexx procedures, press a function key (**F1 though F10**). To define or re-define a function key, press **CONTROL+Function key**, and a window will open containing the text to be changed. To Edit an external script, press **SHIFT+Function key** and the editor will be invoked with that script name as a parameter. To define or re-define the edit invocation, press **CONTROL+ESC**. You can also set the edit invocation from the program's icon so that it will always come up configured the way you prefer. This is what we have done with the release disk; it's simply set up (in the icon) to call 'ed', the screen editor that is supplied with the WorkBench disk.

Macros and Scripts that are invoked from within the AVT system have the port address already set correctly. If you wish to send a command to the AVT software from an ARexx script invoked from the CLI or another ARexx host, such as a database, use this command to do so (case *is* significant here) before attempting to execute an AVT command:

address AVT_Port;

This is not required when a macro or script is invoked from within the AVT system - you don't need it in a macro key, for instance.

Editing Scripts With An External Editor

This is a unique and useful function of the AVT system. If you have an external script, it is of course a separate text file. In the AVT system the means used to invoke such a script is to simply place the filename in a macro key, and when that macro key is pressed, that script is sent to ARexx automatically.

In order to provide a convenient, interactive environment for you to handle your script files, this software has been provided with a means to invoke an editor with any macro key's contents as its file to be edited. The means for this is a "special" macro, the **F11** macro. The AVT system, when started from the WorkBench, configures the **F11** macro in this way:

"arg file; towb; address command; ed file; address; toavt"

When ARexx is presented with a macro that is quoted, like the one above is, if there is any text after the trailing quote it will attempt to interpret that text as a parameter to the macro. Let's assume, for the purposes of this example, that the **F3** key contains the full script name **"rexx:Script.rexx"**.

When you press **SHIFT+F3**, the AVT sends the **F11** macro to the ARexx command processor, but appends the filename in that key to the above macro, so that the last part of it will look like this:

...ed file; address; toavt"rexx:Script.rexx

The first command in the macro (**arg file;**) tells ARexx (a) that the macro does indeed have a parameter, and (b) that wherever the keyword (**file**) is encountered in the macro, to replace it with the parameter.

The final result is this series of commands:

"arg file;
The leading quote indicates the beginning of the macro to Arexx. The actual command informs ARexx of the macro's argument (rexx:Script.rexx) and tells it to replace the occurrence of **file** after the CLI **ed** command with the that argument, as shown below.

towb;
Takes you (the user) to the WorkBench screen so that you are on the screen where the **ed** editor will open the edit window.

address command;
Changes Host to AmigaDOS for CLI command execution. Since **ed** is a CLI command, you must do this to execute it - otherwise, the **ed** command is sent directly to the AVT system, which does not know how to execute it, and this would be an error.

ed rexx:Script.rexx;
Invokes the CBM supplied screen editor on the script, just as if you had typed it on the CLI command line. You can change this line from saying **ed** to **UE** to invoke UEdit, or **CED** to invoke CygnusEd, etc. The next command (**address;**) will not execute until you finish the edit session by quitting the editor, and optionally saving the modified script file.

address;
Restores the AVT_Port as the current Host; otherwise, the **toavt** command that follows would be sent to the CLI, which would not recognize it and would then return an error.

toavt"
Brings you (the user) back to the AVT screens; the closing quote signals ARexx that this is the end of the Macro - in addition, since the first statement indicated there was an argument to this Macro, ARexx begins looking here for the actual argument- the AVT has appended the filename here so that this will work.

AVT ARexx Command Set

This section details the unique commands available within the scope of the AVT system. These commands are specific to the AVT, and should only be used when the Host address has either been specifically set to **AVT_Port** or when a macro or script has been invoked directly from with an AVT macro key.

These commands range from simple general commands that allow you to select any menu item from a macro to unusual things like the ability to change a color register while in the FAX mode - useful for generating pseudo colored images to enhance detail.

ARexx Command List:

WRING

This command places the system in a **"wait for ring"** state. A window is opened on the main control panel that indicates this, and has a close gadget. If you wish to terminate the wait state, press the close gadget.

The command always returns a value to the script or macro that invoked it - if the value is 0, then the phone *is* ringing. If the value is 1, then the user aborted the wait by pressing the window's close gadget. If the value is other than 0 or 1, then there was a system problem such as being unable to open the window itself.

RXIMAGE dly

This command operates in a similar fashion to the AUTO button on the main control panel, the difference being that only one image is received, then the command returns, and that there is a maximum delay of DLY seconds; if an autostart signal is not received within that time, the function aborts, returning the value 1024.

If an image is received, a value between 0 and 127 will be returned. The code returned varies depending on which type of mode it went into. The image is received and displayed on screen before the function returns, if the autostart code is received.

RXCODE dly

This command will wait for reception of a particular code for DLY seconds. If the code is received, it is returned as a number from 0 to 127. If not, 1024 is returned.

This can be used in conjunction with the **SENDCODE** function to provide any arbitrary type of handshaking between units.

SENDCODE value

This command sends a code according to the value you supply it. Valid numbers are 0 to 127.

This can be used in conjunction with the **RXCODE** function to provide any arbitrary type of handshaking between units.

MENU (menu, item, subitem)

This command allows you to execute any menu selection just as if you had selected it with the mouse, by hand. There are three parameters to this command, one for each of the possible levels of menu available. For instance, the **ABOUT...** menu selection, which simply tells you about the program, is in the first menu - this is menu zero. It the very first item in the menu, which is also zero, and there is no subitem associated with it, so the value for that parameter can be anything - though it *must* be included. You would use **MENU 0,0,0;** in a macro or script to select this menu item. Another example: If you wished to be able to send your cwid just by hitting a key, you could place this in a macro key:

```
"MENU 4,2,0;"
```

Whenever that FKEY was pressed, your CW ID would be sent out.

REQUEST (file | text | box)(,|)(control string)

This menu item allows you to use the AVT's internal requesters to obtain information for your own uses. You can bring up a file requester which will return a filename to your script or macro, a text requester which will return a text string to your macro, or a box requester that will return which button (left or right) was pressed by the user.

These requester capabilities allow you to create virtually any kind of program in ARexx which interacts with the user entirely within the AVT environment.

The first option we'll discuss is the text requester. This brings up a text box on the control panel that can either be empty or filled with a preset string you provide. In addition, you provide text that is used as a title to the window the text box is contained in, and text that appears below the text box, generally used for instructions.

Sample syntax for this usage is as follows:

```
REQUEST TEXT old text,window title,instructions;
```


This would cause the text requester to appear on the control panel *containing* the phrase "old text". The window would be titled "window title", and right underneath the text area the word "instructions" would appear.

Whatever text the user leaves in the text box when he closes the window or hits one of the <return> or <enter> keys will be returned to your calling macro.

The next usage of the **REQUEST** function uses the **FILE** keyword. This allows you to bring up the file requester set up in any way you please. You can set the path where it first looks, the filename string, the filename extension, and a name for the "action button" that causes the requester to return to the script with the selected filename.

The specific control string format is as follows:

```
REQUEST FILE path,filename,extension,action_button
```

Where the action button text is limited to eight characters maximum, one character minimum. Observe standard limitations for AmigaDOS filenames, directories, and paths for the other parameters.

When called in this manner, the requester will either return the filename, for example **disk:path/filename.extension**, or it will return the string **FR_CANCELLED**. This indicates that the user hit the cancel button, instead of the "action button". Here's an example of how you might use this:

```
options results;
REQUEST FILE REXX:,,.rexx,Ld Macro;
filename = result;
options;
if filename ~= FR_CANCELLED then do
     /* desired operation on filename here... */
     end;
```

This would invoke the file requester such that it would be set to the REXX: assignment (directory or disk) in your system, there would be no filename preset (none was specified) the extension would be .rexx, and the action button would be labeled **Ld Macro**.

If the user hit the **Ld Macro** button in the requester, you should get a complete filename, including the path, returned to your ARexx macro or script. If the user hits the **CANCEL** button instead, you will receive the string **FR_CANCELLED.**

The last manner in which you can utilize the **REQUEST** operator is with the **BOX** keyword and control string. This configuration allows you to present the user with a standard Intuition-type requester with one or two lines of explanatory text, and two labelled buttons. The user must press one of the two buttons, and the operation simply returns an **L** or an **R**, depending on which of the left or right buttons was pressed by the user.

Here is the control string format for the **BOX** operator:

```
line 1,line 2,Right button text, Left button text;
```

And here is an example of its usage:

```
options results;
REQUEST BOX, press either button,A or B,-A-,-B-;
whichbutton = result;
options;
if whichbutton == L then do
     /* "A" operation here */
     end
else do
     /* "B" operation here */
     end
```

CLOSE

The **CLOSE** operation terminates the AVT program immediately.

SETS1 value

This allows you to set the trim value for the reception of synchronous SSTV in the Scotty S1 mode.

SETS2 value

This allows you to set the trim value for the reception of synchronous SSTV in the Scotty S2 mode.

SETM1 value

This allows you to set the trim value for the reception of synchronous SSTV in the Martin M1 mode.

SETM2 value

This allows you to set the trim value for the reception of synchronous SSTV in the Martin M2 mode.

SET12 value

This allows you to set the trim value for the reception of synchronous SSTV in the Robot 12-second composite mode.

SET24 value

This allows you to set the trim value for the reception of synchronous SSTV in the Robot 24-second composite mode.

SET36 value

This allows you to set the trim value for the reception of synchronous SSTV in the Robot 36-second composite mode.

SET72 value

This allows you to set the trim value for the reception of synchronous SSTV in the Robot 72-second composite mode.

SETTS1 value

This allows you to set the trim value for the transmission of synchronous SSTV in the Scotty S1 mode.

SETTS2 value

This allows you to set the trim value for the transmission of synchronous SSTV in the Scotty S2 mode.

SETTM1 value

This allows you to set the trim value for the transmission of synchronous SSTV in the Martin M1 mode.

SETTM2 value

This allows you to set the trim value for the transmission of synchronous SSTV in the Martin M2 mode.

SPEAK text

This command allows you to transmit any text as synthesized speech with one simple line. For instance, you could use the command like this:

"SPEAK This is Ay Ay 7 Ay S in Glasgow, Montana";

Or...

"SPEAK The frequency is in use! Please Q S Y";

CWSEND text

This is similar to the SPEAK command, except it sends the text in morse code. Usage:

"CWSEND freq is in use om, pse qsy de aa7as"

SENSE

Now *here* is an interesting command. It actually causes the AVT system to read the AVT hardware and returns a period value to you - this is directly related to the current frequency on the input of the AVT hardware.

The following script example uses this in conjunction with the **DLED** command, which displays information in a simulated LED display drawn on the scope area of the control panel.

```
/* readperiod.rexx (this is a comment) */
options results;
'SENSE'; /* reads period from AVT */
pvalue = result;
options;
fvalue = 1/(pvalue * .0000002384185791);
temp = int(fvalue); /* get the integer result */
'DLED' temp;
exit 0

/* subroutine to create integer result */
int:
     arg incoming;
     return(incoming%1);
```


ILED

The **ILED** command initializes the simulated LED display by erasing the area around it, and drawing it with 0 as the readout value.

DLED value

The **DLED** command draws the LED display and sets it to the parameter supplied. See the example script under the **SENSE** command description.

ELED

The **ELED** command removes (erases) the LED display so that the scope simulation again appears complete.

INSET (1|2|3|4|5|6)

This command changes the input selected to the one you choose with the numeric parameter. The numbers one through five correspond directly with the inputs marked on the AVT system's hardware box. Usage:

INSET 1;

You may also use the form **INSET 6**, which will select the telephone interface. It will cause the telephone to be "picked", or become off-hook. At this point you can dial using the **DIAL** command, or hang up again by selecting one of the other five inputs, as in **INSET 1;**

DIAL

This operation causes standard dual tone pairs to be sent accordingly. The function can be used either with the telephone, or on other bands where repeaters, for instance, might use TTP operations to control various functions, or even operate an autopatch. Usage:

'DIAL 1234567890*#ABCD';

PASSTHRU

This command routes the currently selected input through to the telephone jack. It allows you to patch your HF radio, for instance, to the telephone line. You would first select the input, and then use the PASSTHRU command. To turn this off, use the NOTHRU command.

TXTHRU

This command routes the signal from the Amiga to the output jacks, and keys the unit as well. It is essentially the same situation as if the AVT Master software had gone into transmit on its own, except that the software is not driving the audio port. You might use this feature to route music to the telephone, or speech from another piece of software. To turn this off, use the NOTHRU command.

NOTHRU

This command turns off the KEY line, and disables the routing of audio thru the unit. It is used after PASSTHRU or TXTHRU commands have been executed.

TOWB

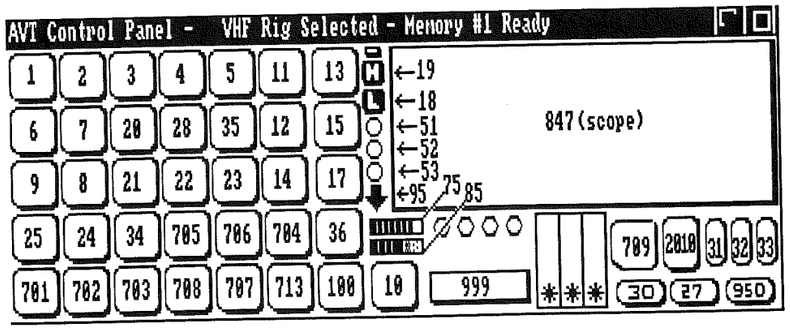
This command causes the WorkBench screen to be brought to the front, hiding the AVT screen(s) behind it. It is useful whenever you have invoked a standard CLI type command from within the AVT system.

TOAVT

This command causes the AVT screen to be brought to the front, hiding any other open screen(s) behind it. It is useful whenever you have invoked a standard CLI type command from within the AVT system and wish to return to the AVT screen when the command completes.

PRESSA gadgetnumber

This allows you to press any button on the main control panel by supplying a gadget number that matches the button. See the following illustration for the numbers.



Gadgets with accompanying numbers

SLOAD filename

This function allows you to load an SSTV image file by name. It does not default to any particular path, nor make any assumptions. You must supply the entire path name. Here is an example of its use that duplicates the SSTV menu's "Load SSTV" function for .high images:

```
options results
'REQUEST FILE ,,.high,Load SSTV'
filename = result
options
if filename ~= FR_CANCELLED then do
     'SLOAD filename'
     end
```

SSAVE filename

This function allows you to save an SSTV image file by name. It does not default to any particular path, nor make any assumptions. You must supply the entire path name. Here is an example of its use that duplicates the SSTV menu's "Save SSTV" function for .high images:

```
options results
'REQUEST FILE ,,.high,Save SSTV'
filename = result
options
if filename ~= FR_CANCELLED then do
     'SSAVE filename'
     end
```

ISAVE filename

This function allows you to save an IFF image file by name. It does not default to any particular path, nor make any assumptions. You must supply the entire path name. Here is an example of its use that duplicates the SSTV menu's "Save IFF" function:

```
options results
'REQUEST FILE ,,.iff,Save IFF'
filename = result
options
if filename ~= FR_CANCELLED then do
     'ISAVE filename'
     end
```

PSEUDO (level,r,g,b)

This function *only* operates when you are in FAX mode. It allows you to set the color registers to a particular value so that you can color the FAX pictures as you wish. There are 16 color registers, each with individual values or red, green and blue. here is an example script that sets them all from a random number you supply:

```
/* color mangulate script */

options results
request "TEXT,,Pseudo Color,Enter seed from 1-9999"
seed = result
options

dummy = random(0,15,seed)

do i = 1 to 13
  red = random(0,15)
  green = random(0,15)
  blue = random(0,15)
  pseudo i || "," || red || "," || green || "," ||
blue
  end
```

ASGREY

This function is provided as a handy way to return all the color registers back to their standard grey levels after a number of PSEUDO coloring operations.

Tutorial

The following is a general tutorial designed to get the program and the hardware up and running, and the operator's "feet wet" with the system. It is *not* a complete, step by step walkthrough of the system - that would take far too much space. Instead, we show you how to receive an image, how to transmit one, how to start and end the program, and a few other simple items. Once you have accomplished these things, the rest is a matter of working with the system and gaining familiarity with it's many capabilities and features.

The first thing that needs to be done is to make sure that all the physical connections are correctly made. We'll go over them quickly.

First, with the Amiga *off*, the supplied shielded parallel cable is attached to both the Amiga's parallel port and the AVT's parallel connector, carefully observing that the cable's connectors are seated squarely against the mating connectors on both the Amiga and the AVT hardware.

NOTE: If you are using a parallel switch, then turn the switch ***now*** to the position that selects the AVT hardware, rather than your printer, digitizer, or whatever else might be connected to the parallel switch.

For radio operators:

Attach an audio cable from your receiver to input number one on the AVT hardware, then attach an audio cable from the **"Transmit Output"** connector on the AVT hardware. You may wish to connect the AVT's keying line to your transmitter as well; this allows the AVT to key your transmitter without requiring that you hold the microphone keyed too.

For those who will be using the telephone only:

Use a **"Y"** telephone connector to connect the AVT hardware to your telephone line, or, if you have a spare jack, just plug it into that jack.

Connect an RCA cable from the Amiga's **left** audio output jack to the **"Amiga In"** jack on the AVT Hardware.

Now plug the power cable from the supplied wall transformer into the AVT hardware, and then plug the transformer into an AC outlet.

Finally, apply power to your Amiga and boot it up using your standard WorkBench disk. This completes the hardware insallation procedure.

NOTE: The WorkBench is a trademark of Commodore Business Machines, Inc.

Starting the Program

For single disk drive owners:

Insert a *copy* of the supplied disk into drive df0: and double-click on the disk icon when it appears. The disk's window will open, and the AVT Master icon will appear within it. Double-click on the AVT Master icon to start the software running. During the program's initialization, it may ask you to re-insert the WorkBench disk. Do so, and if you are prompted to re-insert the AVT disk again, do that as well. These prompts are a result of having only one disk drive, while the program requires data from both your boot disk and its own disk to start. We strongly recommend the use of a second drive if this is possible.

For dual (or more) disk drive owners):

Insert a *copy* of the supplied disk into drive df0: or df1:, whichever is empty. Ensure that the WorkBench disk is also inserted, or the system will demand that you do so anyway. Double-click on the disk icon when it appears. The disk's window will open, and the AVT Master icon will appear within it. Double-click on the AVT Master icon to start the software running.

For hard drive owners:

If you wish to install the AVT software upon your hard drive, please do so later, once you know more about the AVT system. It is not copy protected, however it does have a few ancillary files that also need to be installed for the most effective operation of the system. For now, just insert the disk in any disk drive, and double-click on the disk icon when it appears. The disk's window will open, and the AVT Master icon will appear within it. Double-click on the AVT Master icon to start the software running.

If the hardware has been installed correctly, and you have enough memory (one megabyte, ***minimum***, and preferably 1.5 meg or more) then the AVT system will initialize correctly. If you do not have ARexx installed and running in your Amiga, a requester will inform you of this fact, and indicate that the Macro and Script capabilities are disabled - this is not a problem, except that it is a shame to not have them available. ARexx is an optional addition to the system that you can purchase at your Amiga dealer; refer to the section of the manual on ARexx for details. For now, if you don't know what it is, don't worry about it.

Correct initialization is indicated by the appearance of the control panel without any error requesters with the single exception (possibly) of the ARexx macro warning, which is not an error.

Once the program has started, you should see a control panel at or near the bottom of the screen, with a completely black area visible behind it. The AVT system uses a "control panel" analogy at all times. To operate the control panel, place your mouse pointer over the control or button (often referred to as a *Gadget* in the text of this manual) and press the **left** mouse button. The AVT system also utilizes standard Amiga menus, which are accessed with the **right** mouse button. At no time is the right mouse button used for anything besides menu control.

To receive an image, you need to find someone who can send one to you, or you need to record one on a tape deck so you can play it back into the system. For those who are operating the system over the phone, you must call someone up. For those operating using a radio, you must tune in a station sending images - if you are operating upper or lower side band, you must tune to within 50 cycles of the frequency the other station is operating upon, or the automatic receive function will *not* activate.

For radio operators:

Click on the button marked **I/o**. The I/o Routing control window will open - click on the box marked **"Input 1"**, and then close the window by clicking on the close gadget at the upper left hand corner of the I/o Routing control window.

For those using the telephone:

Click on the button marked **I/o**. The I/o Routing control window will open - click on the box marked **"Telephone"**, and then close the window by clicking on the close gadget at the upper left hand corner of the I/o Routing control window.

Note: This sequence "picks up" the telephone; when you have completed sending your images, you *must* remember to open the I/O window and click on another gadget (any of Input 1 through Input 5) so that the AVT hardware "hangs up".

Once this has been done, click on the button marked **AUTO** near the top left of the main control panel, and request that the other person or station send you an image in any one of the following modes:

Robot 12-, 24-, 36- or 72-second composite color;
Robot 8-, 12-, 24- or 36-second black and white;
AVT 24-, 90- or 94-second color;

Other modes will work also, but these are the most common, and operate fully automatically. When the other person sends the image, the AVT will take it from there. After the image is completely received, it is processed into color in a few seconds, and then displayed for you.

To send this image back, you must first choose a mode to do this in. A good idea is to send it back in the mode it was sent to you originally. You can either ask the other person what that mode was, or you can examine the control panel buttons - they will tell you by which ones have the little red indicator in their upper left hand corner illuminated.

If the other person sent you an AVT mode image, such as 24, 90 or 94, then all you have to do to send it back is press the **TX** button.

If the other person sent you a Robot mode image that was black and white, the same - simply press the **TX** button once.

However, if the other person sent you a Robot Color image, you have to take one extra step. Click on the **"Synthesize"** button to open the Synthesis and Conversion window. Then click on the button that matches the mode you were sent - 12, 24, 36 or 72. The AVT will prepare the image for transmit in a few seconds, and then close the Synthesis and Conversion window automatically. At this point, you can proceed and press the **TX** button. You might care to notice that one of the red indicators directly over the **Synthesize** button is illuminated - this means that the system is ready to transmit in this mode. If one of these is *not* illuminated, the system will not transmit in a Robot Color mode.

That's really all there is to most receiving and transmitting. You have literally hundreds of other features at your fingertips, so you can look forward to many sessions where you'll discover some enjoyable new feature. We suggest a careful and *thorough* reading of this manual; there is simply no other way to learn all about the AVT system.

We wish you the best of luck - and if you're an amateur radio operator, we hope to see you on the air on 14.230 or 14.233!

APPENDIXES

Appendix A - Slow Scan Television Encoding Methods

Grey Level Encoding

SSTV is a pure FM (Frequency Modulation) type of signal. For any portion of an image, a tone is transmitted that is of a frequency that is directly related to the brightness (luma) or coloration (chroma) of that portion.

Classic SSTV modes utilize the audio frequencies from 1500 Hz, which is interpreted as black, through 2300 Hz, which is interpreted as white. Most of the older SSTV units, with the notable exception of the Microcraft units, break the intervening frequency range into 16 discrete intervals. This means that for every 50 Hz difference in the signal, you would be seeing the next grey level. The range from 2300 to 1500 is 800 cycles, and you simply divide this by 16 to determine the number of Hz/step.

More modern SSTV equipment divides this same 800 Hz portion of the spectrum into 64 grey levels. This results in a new grey level every 12.5 Hz, which is near the limits of what can be achieved on the HF bands. This is because frequency shift due to changing propagation can modify the received signal enough to affect the grey level at these precisions.

Sync Encoding

In addition to sending the luma/chroma for a particular area, most SSTV modes also send a **sync** signal. 1200 Hz is typically used to send this sync information, if it is used. Older SSTV equipment required a sync signal because their timing did not run with enough precision to track the signal through the entire image without the image skewing sideways.

Unlike previous SSTV modes, the AVT mode generates (and tracks) SSTV signals that do not use sync at all. Instead, at the beginning of the image a digital header is transmitted that contains information which describes the mode the image is to be sent in, and timing characteristics for that mode. In this way, the AVT system can track an incoming signal all the way through without having to resort to realignment of its timing circuitry or software by a sync signal.

The VIS standard

With the introduction of its 400c, 450c and 1200c scan converter systems, Robot Research began using an automatic image type identification system known as **VIS**. This is a system whereby the vertical sync signal transmitted before the image contains a short digital burst which identifies the mode the image is being sent in. Since sync, for SSTV purposes, is nominally 1200 Hz, Robot decided to encode the digital information as 1100 Hz and 1300 Hz. The resulting somewhat musical sync signal begins with a 1200 Hz "start" bit, then proceeds through seven data bits and one parity bit.

Once Robot published this technique, the two hams in Europe who modified the 1200c's eproms for the "Scotty" and "Martin" modes also used it, naturally enough - the system was already in use in the scan converters they were modifying. They chose previously unused codes within the seven bit limit for their new modes, and this worked fairly well.

Unfortunately, the 1200c, at least, does not fully decode all seven bits, and will start receiving when a new mode is heard - even though it really can't handle it. Perhaps this will be fixed in newer models of the 1200c.

The AVT also uses the VIS system, since it is in place as a de facto standard. Some of the remaining unused codes were chosen to signal the beginning of an AVT mode. For the AVT modes, however, the VIS signal is *not* used as a sync pulse. The VIS signal is far too imprecise to perform that function for an AVT mode. Instead, the VIS is simply utilized as a mode identifier for other scan converters. An AVT mode performs all sync functions utilizing the digital header that precedes each image transmission.

In this way, all modern SSTV modes have one thing in common- the system receiving the mode in question has a means to determine if it has the capability to receive a particular mode, and if so, to automatically start the receive process.

Appendix B - Ham Slow Scan Operating Frequencies

80 Meters	03.845 MHz, LSB	Advanced Portion
40 Meters	07.171 MHz, LSB	General Portion
20 Meters	14.230 MHz, USB	General Portion
	14.233 MHz, USB	General Portion (Amiga Group)
15 Meters	21.345 MHz, USB	General Portion
10 Meters	28.680 MHz, USB	General Portion

Appendix C - FAX Signals Of Interest As Of Mid-1989

FM FAX - Basic AVT System Compatible

20.737.88 MHz - LSB - 60 LPM - Newsfax - Worldwide
8.078.00 MHz - USB - 120 LPM - Weatherfax, Norfolk, Va.
10.863.00 MHz - USB - 120 LPM - Weatherfax, Norfolk, Va.

AM FAX - Requires Overview Systems PCB for reception

137.300 MHz - FM 30 KHz deviation - Soviet WX satellites
137.400 MHz - FM 30 KHz deviation - Soviet WX satellites
137.500 MHz - FM 30 KHz deviation - US (NOAH) WX satellite
137.620 MHz - FM 30 KHz deviation - US (NOAH) WX satellite
137.850 MHz - FM 30 KHz deviation - Soviet WX satellites

AM FAX on TVRO systems (satellite TV systems) - OverView Only

1925.09 KHz - USB - Spacenet 3 (S3), Transponder 17
This is a GOES East retransmission, extremely high quality reception with a normal communications receiver. Connect the *video output* from your satellite receiver to the antenna input of your general coverage receiver, tune to 1925.09 Khz, and enjoy! That is *not* a mistake - the **video** output. Both 120 lpm and 240 lpm transmissions are found here.

Appendix D - Ham FAX Operating Frequencies

14.233.00 MHz - USB - 120 LPM
14.240.00 MHz - USB - 120 LPM

Appendix E - SSTV/FAX Mode Specifications

AVT 24-second color
- Resolution: 128x120, 18 bits (256,000 color)
- Frame Time: 24
- HSV Encode: RGB line sequential and odd/even line sets
- Sync Types: VIS, digital header, synchronous
- Originated: Black Belt Systems, AA7AS

AVT 90-second color
- Resolution: 256x240, 18 bits (256,000 color)
- Frame Time: 90
- HSV Encode: RGB line sequential, and odd/even line sets
- Sync Types: VIS, digital header, synchronous
- Originated: Black Belt Systems, AA7AS

AVT 94-second color
- Resolution: 320x200, 18 bits (256,000 color)
- Frame Time: 94
- HSV Encode: RGB line sequential and odd/even line sets
- Sync Types: VIS, digital header, synchronous
- Originated: Black Belt Systems, AA7AS

AVT 188-second color
- Resolution: 320x400, 18 bits (256,000 color)
- Frame Time: 188
- HSV Encode: RGB line sequential and odd/even line sets
- Sync Types: VIS, digital header, synchronous
- Originated: Black Belt Systems, AA7AS

AVT 125-second color
- Resolution: 640x400, 4 bits (16 grey level)
- Frame Time: 125
- HSV Encode: Luma only
- Sync Types: VIS, digital header, synchronous
- Originated: Black Belt Systems, AA7AS

AVT QRM submode odd/even line set encoding
- Resolution: identical to source mode
- Frame Time: identical to source mode
- HSV Encode: even lines first, then odd lines, RGB sequential
- Sync Types: identical to source mode except for QRM keying
- Originated: Black Belt Systems, AA7AS

Pseudo Visi-tel 5-second B&W
- Resolution: 96x96, 16 grey level
- Frame Time: 5
- HSV Encode: luma only
- Sync Types: H & V
- Originated: Black Belt Systems, AA7AS

Robot 12-second composite color
- Resolution: 128x120
- Frame Time: 12
- HSV Encode: luma/chroma
- Sync Types: VIS, H & V
- Originated: Robot Research, Inc.

Robot 24-second composite color
- Resolution: 128x120
- Frame Time: 24
- HSV Encode: luma/chroma
- Sync Types: VIS, H & V
- Originated: Robot Research, Inc.

Robot 8-second B&W
- Resolution: 128x120
- Frame Time: 8
- HSV Encode: luma only
- Sync Types: VIS, H & V
- Originated: Robot Research, Inc.

Robot 12-second B&W
- Resolution: 128x120
- Frame Time: 12
- HSV Encode: luma only
- Sync Types: VIS, H & V
- Originated: Robot Research, Inc.

Robot 36-second composite color
- Resolution: 256x240
- Frame Time: 12
- HSV Encode: luma/chroma
- Sync Types: VIS, H & V
- Originated: Robot Research, Inc.

Robot 72-second composite color
- Resolution: 256x240
- Frame Time: 12
- HSV Encode: luma/chroma
- Sync Types: VIS, H & V
- Originated: Robot Research, Inc.

Robot 24-second B&W
- Resolution: 256x240
- Frame Time: 24
- HSV Encode: luma only
- Sync Types: VIS, H & V
- Originated: Robot Research, Inc.

Robot 36-second B&W
- Resolution: 256x240
- Frame Time: 36
- HSV Encode: luma only
- Sync Types: VIS, H & V
- Originated: Robot Research, Inc.

Microcraft 17-second B&W
- Resolution: 256x128
- Frame Time: 17
- HSV Encode: Luma only
- Sync Types: H & V
- Originated: Microcraft Corporation

Microcraft 34-second B&W
- Resolution: 256x256
- Frame Time: 34
- HSV Encode: Luma only
- Sync Types: H & V
- Originated: Microcraft Corporation

WA7WOD 25.5-second line sequential color
- Resolution: 128x128, 12 bits (4096 color)
- Frame Time: 25.5
- HSV Encode: RGB line sequential
- Sync Types: H,H,H & V
- Originated: WA7WOD

WA7WOD 51-second line sequential color
- Resolution: 128x256, 12 bits (4096 color)
- Frame Time: 25.5
- HSV Encode: RGB line sequential
- Sync Types: H,H,H & V
- Originated: WA7WOD

WA7WOD 102-second line sequential color
- Resolution: 256x256, 12 bits (4096 color)
- Frame Time: 25.5
- HSV Encode: RGB line sequential
- Sync Types: H,H,H & V
- Originated: WA7WOD

WA7WOD 17-second B&W
Resolution: 128x256, 4 bits (16 grey levels)
Frame Time: 17
HSV Encode: luma only
Sync Types: H & V
Originated: WA7WOD

Volker-Wrasse 24-second line sequential color
Resolution: 128x120, 12 bits (4096 color)
Frame Time: 24
HSV Encode: line sequential RGB
Sync Types: VIS, H & V
Originated: Volker-Wrasse corp, Germany

Volker-Wrasse 48-second line sequential color
Resolution: 128x240, 12 bits (4096 color)
Frame Time: 48
HSV Encode: line sequential RGB
Sync Types: VIS, H & V
Originated: Volker-Wrasse corp, Germany

Volker-Wrasse 96-second line sequential color
Resolution: 256x240, 12 bits (4096 color)
Frame Time: 96
HSV Encode: line sequential RGB
Sync Types: VIS, H & V
Originated: Volker-Wrasse corp, Germany

Volker-Wrasse 16-second B&W
Resolution: 128x240, 4 bit (16 grey level)
Frame Time: 16
HSV Encode: luma only
Sync Types: VIS, H & V
Originated: Volker-Wrasse corp, Germany

Volker-Wrasse 32-second B&W
Resolution: 128x128, 12 bits (4096 color)
Frame Time: 25.5
HSV Encode: RGB line sequential
Sync Types: H,H,H & V
Originated: WA7WOD

Scotty S1 mode line sequential color
Resolution: 128x240
Frame Time: ...
HSV Encode: RGB line sequential
Sync Types: H & V
Originated: Modified Robot ROM code in Scotland

Scotty S2 mode line sequential color
Resolution: 256x240
Frame Time: ...
HSV Encode: RGB line sequential
Sync Types: H & V
Originated: Modified Robot ROM code in Scotland

Martin M1 mode line sequential color
Resolution: 128x240
Frame Time: ...
HSV Encode: RGB line sequential
Sync Types: H & V
Originated: Modified Robot ROM code in England

Martin M2 mode line sequential color
Resolution: 256x240
Frame Time: ...
HSV Encode: RGB line sequential
Sync Types: H & V
Originated: Modified Robot ROM code in England

Appendix F - SSTV Mode to AVT Format compatibility table

24 AVT:

Robot 12-second composite color
Robot 24-second composite color
Robot 8-second B&W (three images)
Robot 12-second B&W (three images)
25.5-second line sequential color
Volker-Wrasse 24-second line sequential color
Pseudo Visi-tel 5-second B&W (three images)

90 AVT:

Robot 36-second composite color
Robot 72-second composite color
Robot 24-second B&W (three images)
Robot 36-second B&W (three images)
Microcraft 17-second B&W (three images)
Microcraft 34-second B&W (three images)
WA7WOD 17-second B&W (three images)
Volker-Wrasse 48-second line sequential color
Volker-Wrasse 96-second line sequential color
Volker-Wrasse 16-second B&W (three images)
Volker-Wrasse 32-second B&W (three images)
Scotty S1 mode line sequential color
Scotty S2 mode line sequential color
Martin M1 mode line sequential color
Martin M2 mode line sequential color

Appendix G - Overview Systems

Overview Systems manufactures and sells an add-on board for the AVT system that converts AM FAX to FM FAX. This is an absolute requirement for those who wish to receive FAX signals direct from satellite sources, such as the NOAH, COSMOS, and GEOS satellite series.

This unit (the "Overview") is a small PCB providing the following functions:

Features:

* Phase locked Sync clock;

* Ability to record and playback satellite passes on an inexpensive stereo tape recorder. Video information is recorded on one channel and phase locked sync clock on the other;

* Automatic detection of satellites as they pass overhead;

* Automatic start/stop of the tape deck, if desired (may require internal connections to the tape deck);

* Automatic start/stop of a scanning receiver, if used (will require internal connections to the scanner);

* Tape recording features can be used to tape AVT synchronous SSTV modes, HF Weatherfax, and HF newsfax (previously not possible due to the synchronous nature of these modes);

* Controls the tape deck *and* the scanner independently of the tape deck. This means that the Amiga need not be powered up during satellite passes. This is especially helpful if your antenna installation is minimal and you have RFI problems during satellite reception;

* A level control that can be panel mounted is also provided so that received signal brightness can be independently adjusted of the receiver volume control;

* Two LED indicators; One indicates the presence of a satellite's 2.4 kHz subcarrier, and the other indicates that the sync clock is phase locked;

* Provided assembled and calibrated, as a single PCB.

Write:

Overview Systems
P.O. Box 130014
Sunrise, Florida
33313

Or Call:

Tim Heffield, N4IFP
at (305) 748-8315

Appendix H - AVT Hardware Alignment

NOTE: This is an adjustment that should only be performed by an individual that feels totally at home with electronic equipment. Incorrect adjustment may lead to complete (temporary for the duration of the maladjustment) malfunction of the AVT hardware, and adjustment without the proper equipment will invariably lead to worse problems of the same type that were initially experienced. If you are not comfortable around complex electronics, do *not* attempt this adjustment!

When the AVT system leaves the factory, it is correctly adjusted. However, over time, this adjustment may drift. If this occurs, you will notice it primarily in two types of operation - FAX operations and AVT QRM operation.

When operating FAX, it will manifest itself as a sideways drift of the incoming (or transmitted) image to one side or the other... the image will appear to bend or slide sideways as it comes down the screen.

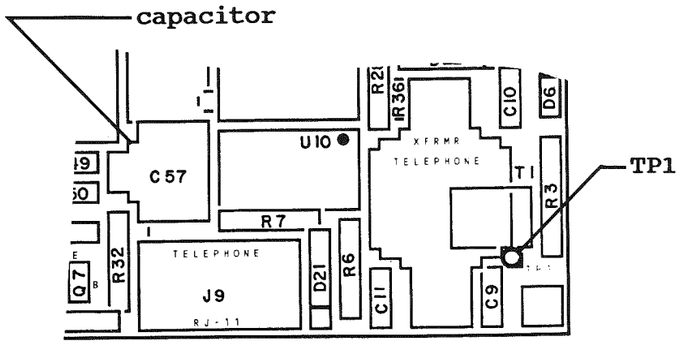
When operating AVT QRM mode, the image will tend to have every other line offset so that the picture has a "jaggedness" to it, sort of like a sawtooth had been pushed into the edge of the picture.

The cause for this type of problem, *if it is consistent*, is the frequency of the master oscillator on the AVT hardware has drifted from its factory setting. If you only see this on a few images, it is most likely that it is the **other** AVT system's oscillator that is off - so until you have verified that this problem is occurring on *all* your images, don't go making any adjustments.

The adjustment itself is simple, and requires that you have a frequency counter that is accurate to one Hz when counting frequencies in the four MHz range. If you have a less accurate frequency counter, you cannot adjust the AVT hardware successfully except by accident.

Remove the top of the AVT hardware case and place it to one side. It is not necessary for the Amiga to be running the AVT master program, nor for any cables (other than the power cable) to be attached to the AVT board to make this adjustment.

Attach the ground connector of the frequency counter to the metal shell of any one of the printed circuit board mounted RCA jacks, and connect the frequency counter's probe tip to TP1, located just to the lower right of the transformer at the lower right corner of the board, when you hold the board with the components facing up, and the side with the two RCA jacks towards you.



PCB layout, TP1, trimmer capacitor locations

Adjust the trimmer capacitor until the frequency counter reads 4.194304 MHz, within about 10 cycles. Even though the adjustment is +/- 10 cycles, you still must use a 1 Hz accuracy frequency counter, because otherwise you don't *know* if you're within ten cycles! Again, the acceptable reading is anywhere between 4.194294 and 4.194314 MHz. Make the adjustment after the AVT has had at least 10 minutes to "warm up", as this allows the components to become temperature stabilized to some degree. Use a non-metallic adjustment tool, or one that has only a metal tip in a plastic or epoxy shaft so that the presence of the adjustment tool does not affect the oscillator's frequency of operation.

You will find that the trimmer capacitor is extremely sensitive, and that the adjustment to within specification takes a very steady hand and a little bit of patience. This is normal.

Appendix I - Program Startup, WorkBench And CLI Options

The AVT program offers a variety of startup options that may be utilized either from the WorkBench or the CLI.

You can always get a list of these options when you are in the CLI mode by typing:

AVT_Master ? <cr>

Which produces a list of options similar to this one:

CLI Parameters:

AVT_Master [ypo] [val] [wfax] [nfax] [gfax] [start] [phase] [version] [vw] [fifty] [chip] [lp] [?qrm] [id] [cw] [compress] [supress] [female] [robo] [deep] [high] [slow] [quick] [squash] [path <SSTV:raw/path>] [ipath <SSTV:iff/path>] [fpath <FAX:iff/path>] [mem <val>] [vcall <text>] [call <text>] [ch <val>] [level <val>] [vlevel <val>] [tlevel <val>] [clevel <val>]

WorkBench ToolTypes:

PARMS=id|cw|version|gfax|wfax|nfax|phase|start|qrm|vw|fifty|lp|female|robo|chip|deep|high|slow|quick|squash|compress|supress

PATH=SSTV:raw/path
IPATH=SSTV:iff/path
FPATH=FAX:iff/path
CALL=CW ID text information
VCALL=Voice ID text information
LEVEL=value (0 -> 63) Audio gain level, SSTV
VLEVEL=value (0 -> 63) Audio gain level, Voice
CLEVEL=value (0 -> 63) Audio gain level, Cw ID
TLEVEL=value (0 -> 63) Audio gain level, Touch Tone Pad
CHAN=value (1 -> 4) Audio Channel
MEM=value (1 -> 16) Number of memories

The notation used for the CLI options ([], <>) is defined as follows: a term surrounded by the square brackets is an option, plain and simple. You can use it or not, as you see fit according to your needs. The [version] term is an example of this.

A term surrounded by the angle brackets MUST be included. In this version of the program, only terms that go along with options are absolutely required. For instance, the term [mem] is an option - but if you look at the list of options printed by the program, you will see that it isn't written that way - it's written: " [mem <val>] ", which means this:

You can type "mem" or not, as you choose. But if you DO choose to type mem, you MUST supply a number from one to 16 to go along with it.

In the future, if there is a required parameter, it will be in the option list as " <option> " without and square brackets surrounding it.

NOTE: When using the WorkBench ToolTypes to select the program's startup options, you may experience problems such as the various string getting "scrambled" in such a way that you cannot fix the them again. This was a bug in the WorkBench itself, and is present in WorkBench 1.2 and 1.3 as well. It has been fixed in WorkBench 1.3.2, which can be obtained from Commodore, from your dealer, or downloaded from Compuserve if you have modem capability. If you are using 1.2 or 1.3, experience has shown that the best way to avoid this problem is to move to the *last* item in the tooltypes list in the icon *before* you hit the **save** button. We regret any inconvenience this bug may cause you, but recommend that you obtain the 1.3.2 WorkBench revision at your earliest opportunity.

Here are the details on the specific options available:

CLI: **[path <sstv:raw/path>]**
WB: **PATH=sstv:raw/path**

If you type:

AVT_Master path piclib <cr>

The program will look for it's "raw" SSTV files in a directory called "piclib" - and it will expect to find that directory in the current directory. If you type:

AVT_Master path df1:piclib <cr>

It will expect to find the "piclib" directory on the DF1: drive. Use this option to cause the program to search where you usually keep you SSTV files.

CLI: **[ipath <SSTV:iff/path>]**
WB: **IPATH=SSTV:iff/path**

If you type:

AVT_Master ipath ifflib <cr>

The program will save "IFF" SSTV files in a directory called "ifflib" - and it will expect to find that directory in the current directory. If you type:

AVT_Master path df1:ifflib <cr>

It will expect to find the "ifflib" directory on the DF1: drive. Use this option to make the program search where you usually keep you IFF files.

CLI: **[fpath <FAX:iff/path>]**
WB: **FPATH=FAX:iff/path**

If you type:

AVT_Master ipath ifflib <cr>

The program will save "IFF" facsimile files in a directory called "ifflib" - and it will expect to find that directory in the current directory. If you type:

AVT_Master path df1:ifflib <cr>

It will expect to find the "ifflib" directory on the DF1: drive. Use this option to make the program search where you normally keep your FAX files. Keep in mind that FAX files are *huge*, on the close order of 600k each. This means that only one FAX image can fit on a standard 880k floppy.

CLI: **[mem <1-16>]**
WB: **MEM=1**

If you type:

AVT_Master mem 2 <cr>

The program will attempt to allocate memory for TWO full function SSTV image memories. If there is not enough memory, you will only get one; likewise, if you typed:

AVT_Master mem 16 <cr>

...and there was not enough memory for 16 image buffers, you would only get as many as there was memory for. It takes 192k per image buffer - so for all 16, you need just over three megabytes of memory on top of what the program itself requires. We do reccomend that you do *not* use all of the Amiga's available memory for image buffers, because that will not allow you to run other programs at the same time, which can be inconvenient.

CLI: [vcall <Voice ID text string>]
WB: VCALL=Voice ID text string

This option allows you to pre-set your call sign information for the automatic Voice ID function. You cannot use spaces when setting this from the CLI, because this "breaks up" the key words for the program. Instead, use one of these "equals" characters: "=", which will not be spoken. They are converted to spaces by the program.

CLI: [call <CW ID text string>]
WB: CALL=CW ID text string

This option allows you to pre-set your callsign information for the automatic CW ID function. You cannot use spaces when setting this from the CLI, because this "breaks up" the key words for the program. Instead, use one of these "equals" characters: "=", which will not be sent as CW. They are converted to spaces by the program.

CLI: [cw]
WB: PARMS=cw

This option, used with the above callsign option, sets the system up for use of a CW ID, rather than the default Voice option.

CLI: [squash]
WB: PARMS=squash

This option causes the control panel to be rendered in interlace mode. The effect of this is to make the control panel 1/2 its original height, in effect "squashing" it. It helps to show more of the actual SSTV and FAX images on screen at one time.

CLI: [compress]
WB: PARMS=compress

This option causes the program to save images with color detail set to 4,096 colors, instead of the 256,000 colors that is the default. For Amiga users, this allows you to have twice as many images in the same amount of disk space, and since a stock Amiga can only display 4096 colors anyway, you won't be able to tell the difference. Owners of other scan converters, such as the Robot, may be able to tell, however.

CLI: [supress]
WB: PARMS=supress

This option causes the program to suppress all text output that is associated with the normal startup, such as the program name, it's version, etc. This allows the program to run without writing any text to your CLI window. Notice that there is only one "p" in the parameter.

CLI: [female]
WB: PARMS=female

This option starts the system with the voice characteristics set for a "female" voice. Since the voice is synthesized, you may find that if you do not also modify the pitch and rates for "female" type characteristics, it will sound.... er... strange. The default is a "male" voice.

CLI: [slow]
WB: PARMS=slow

This causes the speech to be spoken slowly. The default is a medium speed.

CLI: [fast]
WB: PARMS=fast

This causes the speech to be spoken quickly - the default is a medium speed.

CLI: [deep]
WB: PARMS=deep

This causes the speech to use a deep voice. The default is an average depth (for a male) voice.

CLI: [high]
WB: PARMS=high

This causes the speech to use a high voice. The default is an average depth (for a male) voice.

CLI: [robo]
WB: PARMS=robo

This causes the speech to use a non-inflected voice, which sounds somewhat "robotic". The default is an accurately inflected voice, referred to, somewhat tongue in cheek in this writer's opinion, as a "natural" voice.

CLI: [id]
WB: PARMS=cw

This option, used with the above ID and CW options, causes the system to start up in a mode such that after every transmission, the ID will be sent as specified by the CW and voice options.

CLI: [version]
WB: PARMS=version

This option acts differently in the CLI than it does in the WB startup. In the CLI, if you include this in the command line, the program will print the version number and then immediately terminate.

If the version keyword is used in the ToolTypes in the icon for the WB startup, the program will print the version in the small window it opens on the WB and then start normally.

Use this option when writing AEA in reference to any aspect of the system's performance - we need it to understand the situation fully.

CLI: [qrm]
WB: PARMS=qrm

This option instructs the AVT system to initialize the QRM defeat processing to ON. The default is off.

CLI: [chip]
WB: PARMS=chip

This option **must** be used by those who have Amiga 500's or 2000's with only one megabyte of RAM. This option enables special screen managment routines within the AVT software, without which many of the SSTV and FAX modes will not function properly.

The specific problem that the "chip" option was developed to address is that of DMA contention between the Amiga's screen display hardware and the CPU in a machine that has no memory that is devoted to the CPU only (this is called "fast" memory in Amiga terms).

During receive situations, when the program is invoked with the chip option, a "ghost" screen that is a black and white screen is brought to the front. This is not a HAM (Hold-And-Modify) screen, but a standard four bitplane Amiga color register-based screen. These special screen manipulations ensure that the CPU has full access to the system's memory, and so has enough "horsepower" to handle the data that is coming in at an extremely high rate of speed from the AVT interface.

CLI: [ch <value 1 to 4>]
WB: CHAN=value (1 to 4)

This option allows you to specify, on startup, which of the Amiga's four audio output channels is to be used on transmit for all modes. The setting can be changed at any time once the program is started by simply opening the SET window and changing the selected channel there. This does allow you to set the default, and is especially useful when starting the program from its icon. The default, if no option is specified, is Channel one. For your information, channels one and two are LEFT channels, and three and four are RIGHT channels.

CLI: [level <value 0 to 63>]
WB: LEVEL=value (0 to 63)

This option allows you to specify, on startup, what the output level of the SSTV signal is when transmitting from the Amiga. The setting can be changed at any time once the program is started by simply opening the SET window and changing the selected level shown there. This does allow you to set the default, and is especially useful when starting the program. The default, if no option is specified, is 32, the middle setting.

CLI: [vlevel <value 0 to 63>]
WB: VLEVEL=value (0 to 63)

This option allows you to specify, on startup, what the output level of the Voice ID signal is when transmitting from the Amiga. The setting can be changed at any time once the program is started by simply opening the SET window and changing the selected level shown there. This does allow you to set the default, and is especially useful when starting the program. The default, if no option is specified, is 32, the middle setting.

CLI: [clevel <value 0 to 63>]
WB: CLEVEL=value (0 to 63)

This option allows you to specify, on startup, what the output level of the CW ID signal is when transmitting from the Amiga. The setting can be changed at any time once the program is started by simply opening the SET window and changing the selected level shown there. This does allow you to set the default, and is especially useful when starting the program. The default, if no option is specified, is 32, the middle setting.

CLI: [tlevel <value 0 to 63>]
WB: TLEVEL=value (0 to 63)

This option allows you to specify, on startup, what the output level of the touch tone pad signal is when transmitting from the Amiga. The setting can be changed at any time once the program is started by simply opening the SET window and changing the selected level shown there. This does allow you to set the default, and is especially useful when starting the program. The default, if no option is specified, is 32, the middle setting.

CLI : [wfax]
WB: PARMS=wfax

This option causes the program to immediately enter the weather FAX operations mode (120 lpm) upon startup.

CLI : [gfax]
WB: PARMS=wfax

This option causes the program to immediately enter the GeoFAX operations mode (240 lpm) upon startup.

CLI : [nfax]
WB: PARMS=wfax

This option causes the program to immediately enter the news FAX operations mode (60 lpm) upon startup.

CLI: [phase]
WB: PARMS=phase

This option is used with the wfax and nfax options. It is used when the start option is used to force the system into FAX receive, but at the same time, to bypass the "start" signal and go directly into the phasing state. This is useful for use with stations that do not send start signals, such as many newsfax stations.

CLI: [start]
WB: PARMS=start

This option is used with the wfax and nfax options. It is used to force the system directly into receive mode in the selected FAX line rate. Only ONE image will be received, then the system will exit receive and return to manual mode. See the FAX_Operations file for more information on the use of this option.

A Final Note On The WorkBench PARMS Command:

You may have noticed that there are a number of terms you can use with the **PARMS=** tooltype. You can use one or more of these terms by typing a vertical bar (found at the upper right of your keyboard) between them:

PARMS=version|chip|wfax|start

Take a look in the supplied ICON if you'd like to see an example of this.

Your Preferences Settings

The mouse pointer is used by the program to indicate that the AVT system is busy at times. In addition, the mouse pointer is used as a "header acquired" indicator when the AVT receive modes are utilized, and during FAX receive and image grabbing.

This is done by "lighting" a section of the pointer when the header is validated by the decryption routines. The mechanics of this are not particularly important, but the colors you pick for your pointer are.

The default color set that comes with the computer is:

1) This is a "clear" color - you don't need to worry about it
2) Black
3) Red
4) White

These colors are convenient - the pointer in the AVT system uses color four (white) to spell out the word busy.

A rectangular area (the "light") is drawn below the word BUSY, also in color four (white). Inside this rectangular area, we place color three (red) at times when the system detects sync.

The net effect of this is that the pointer turns into the word "busy", with an indicator under it that is blank until the header is acquired, and then turns bright red.

If you change the pointer colors, you may make it quite difficult to see this indicator, and that's the point we are driving at.

The supplied preferences pointer colors are very convenient for the uses the AVT system makes of the pointer - you probably want to leave them that way, at least initially.

Appendix J - Image Video Input Devices For The Amiga

All entries have been verified with the manufacturers as of August 1989.

Device: (01) Digi View Gold
Manufacturer: NewTek, 115 West Crane Street, Topeka, KS, 66603. (913) 354-1146
Classification: Digitizer, Monochrome/RGB
Retail Price: $199.95
Capture Duration: Approximately 30 seconds for full color
Accuracy: 21 bits (RGB 7/7/7)

Input is standard NTSC signal. Output to Amiga is via the parallel port. Supports resolutions to 768x480. Discounters typically have the unit available in the range of $120.00 complete. This unit is highly recommended as an optimum compromise between digitizing time and quality. Produces extremely high quality images, sophisticated support software.

Device: (02) a1000 Live!
Manufacturer: A-Squared Distributions, Inc, 6114 LaSalle Ave, Oakland, CA, 94611. (415) 339-0339
Classification: Frame Grabber, Color
Retail Price: $295.00 500-399 2000-450
Capture Duration: 1/4th of a second (15 fields, 7 frames)
Accuracy: 12 bits (RGB 4/4/4)

Input is via standard NTSC composite signal. Output to Amiga is via the system bus (plug in card). Supports resolutions to 320x480. Discounters typically have the unit available in the range of $230.00 See also the a500 and a2000/2500 versions of this grabber.

Device: (03) Frame Grabber
Manufacturer: Progressive Peripherals & Software, 464 Kalmath St, Denver, CO, 80204 (303) 825-4144
Classification: Frame Grabber, Color
Retail Price: $699.95
Capture Duration: 1/30th of a second (2 fields, 1 frame)
Accuracy: 12 bits (RGB 4/4/4)

Input is via standard NTSC composite video. Output to Amiga is via parallel port. Supports resolutions to 384x240. Discounters typically have the unit available in the price range of $540.00

Device: (04) Frame Capture
Manufacturer: Mimetics Corporation, POB 1560, Cupertino CA, 95015. (408) 741-0117
Classification: Frame Grabber/buffer, Color
Retail Price: $199.95/buffer is 549.95
Capture Duration: 1/15th of a second (4 fields, 2 frames)
Accuracy: 24 bits (RGB 8/8/8)

Input is via NTSC video. Output to Amiga is via system bus. Supports resolutions to 640x400. Discounters typically have the unit available in the price range of $160.00

Device: (05) VD1
Manufacturer: Impulse Inc, 6870 Shingle Creek Parkway #112, Minneapolis, MN, 55430. (612) 566-0221
Classification: Grabber/buffer, Color
Retail Price: $1000.00
Capture Duration: 1/30th of a second (2 field, 1 frame)
Accuracy: 24 bits (Composite 8 bits)

Input is via NTSC video. Output to Amiga is via parallel port. Supports resolutions to 352x440. Discounters typically have the unit available in the price range of $800.00

Device: (06) Perfect Vision
Manufacturer: Sunrise Industries, pob 1453, College Station, TX, 77841. (409) 846-1311
Classification: Grabber, Monochrome
Retail Price: $249.95
Capture Duration: 1/60th of a second (1 field, 1/2 frame)
Accuracy: 12 bits (RGB 4/4/4)

Input is via NTSC video. Output to Amiga is via parallel port. Supports resolutions to 320x400. Discounters typically have the unit available in the price range of $175.00. Sunrise Industries also manufactures the "Color Splitter". This is a unit that electronically separates the R, G, and B components of a composite signal, and (when used with the Perfect Vision) automates RGB frame grabbing so it is extremely fast, although still not "real-time" like a color frame grabber. Can be used with DigiView Gold from NewTek.

Device: (07) Professional Scanlab
Manufacturer: ASDG Incorporated, 925 Stewart St, Madison, WI, 53713. (608) 273-6585
Classification: Scanner, Color
Retail Price: $995.00
Capture Duration: Approximately 1 minute, variable
Accuracy: 24 bits (RGB 8/8/8)

Input is via Sharp JX-450 scanned document. Output to Amiga is via IEE-488 bus (included). Supports resolutions to 5100x3300. Discounts available between 5 and 10% from dealers. Extreme high end unit, quality surpasses all other options by miles. The JX-450 is listed for 6995.00, another option is JX-300, which is only(!) 3995.00

Device: (08) Scanlab 100
Manufacturer: ASDG Incorporated, 925 Stewart St, Madison, WI, 53713. (608) 273-6585
Classification: Scanner, Color
Retail Price: $995.95
Capture Duration: Approximately 3-5 minutes
Accuracy: 18 bits (RGB 6/6/6)

Input is via Sharp JX-100 scanned document. Output to Amiga is via serial port (included). Supports resolutions to 200 dpi, square. The unit is not currently available discounted, although it will be available thru dealers so discounts are expected to be available. This is a complete scanning system for 995.00 including the JX-100.

Device: (09) a500 Live!
Manufacturer: A-Squared Distributions, Inc, 6114 LaSalle Ave, Oakland, CA, 94611. (415) 339-0339
Classification: Frame Grabber, Color
Retail Price: $399.00
Capture Duration: 1/4th of a second (15 fields, 7 frames)
Accuracy: 12 bits (RGB 4/4/4)

Input is via standard NTSC composite signal. Output to Amiga is via the system bus (plug in card). Supports resolutions to 320x480. Discounters typically have the unit available in the range of $320.00 See also the a1000 and a2000/2500 units.

Device: (10) a2000/2500 Live!
Manufacturer: A-Squared Distributions, Inc, 6114 LaSalle Ave, Oakland, CA, 94611. (415) 339-0339
Classification: Frame Grabber, Color
Retail Price: $450.00
Capture Duration: 1/4th of a second (15 fields, 7 frames)
Accuracy: 12 bits (RGB 4/4/4)

Input is via standard NTSC composite signal or direct RGB input. Output to Amiga is via the system bus (plug in card). Supports resolutions to 768x480. Dual video inputs; Discounters typically have the unit available in the range of $360.00 See also the a500 and a1000 units.

Device: (11) Frame Grabber 256
Manufacturer: Progressive Peripherals & Software, 464 Kalmath St, Denver, CO, 80204 (303) 825-4144
Classification: Frame Grabber, Monochrome
Retail Price: $724.95
Capture Duration: 1/60th of a second (1 fields, 1/2 frame)
Accuracy: 24 bits (RGB 8/8/8)

Input is via standard NTSC composite video. Output to Amiga is via parallel port. Supports resolutions to 768x480. Discounters typically have the unit available in the price range of $580.00

Device: (12) Spectrum
Manufacturer: Preferred Marketing Services, 1846 Rosemeade Parkway, Suite #148, Carrolton, TX, 75007. (214) 394-4503 - FAX (214) 394-4504
Classification: Digitizer, Monochrome
Retail Price: $199.95
Capture Duration: Approximately 30 seconds for color
Accuracy: 21 bits (RGB 7/7/7)

Input is via NTSC video. Output to Amiga is via parallel port. Supports resolutions to 768x440. Discounters typically have the unit available in the price range of $120.00

Appendix K - About the AVT, and IBM (and its clones)

Many people have looked at the AVT system and immediately asked, "is it available for the PC (meaning the IBM PC and it's clones)". The answer is no, unfortunately. This was not an arbitrary decision - here are some of the underlying reasons why.

One thing that stands out immediately is that the various PCs are very different from one another in many ways. Various amounts of memory from 512k to several megabytes may be available; CPUs may be InTel Corp. 8088's, 8086's, 80188's, 80286's, 80386's, or other manufacturer's higher-performance replacements for these chips. Display capabilities vary from 4 colors to 256 colors. They may have real parallel ports, or they may have the TTL pseudo parallel port that is in the original IBM design. Some run Microsofts MS-DOS, some run IBMs. Some use the Pheonix BIOS, some use IBM's, some use... you get the idea.

This certainly makes things much more formidable - it's not enough to stop someone from doing a project like the AVT, but it sure makes you think carefully before starting.

Display Quality - The Amiga comes stock with the capability to display 4,096 colors at resolutions up to 372x480. This is a good selection for images. To get 256 colors on a PC, you have to spend many hundreds of dollars extra - remember, this would be part of *your* cost to get the AVT up and running, and that with only 256 colors, which would look more like a cartoon than a smooth image in many cases.

Audio Output - The Amiga has 256 level audio resolution with sampling accuracy to approximately 14 khz, which is run under DMA by one of the three co-processors, leaving the CPU free to do other jobs. The PCs have only a single bit (2 level resolution) that can be toggled. Either we would have to design the audio output into the PC-version of the AVT or you would have to purchase it - again, more dollars.

CPU horsepower - the basic, bottom of the line Amiga (the a500) comes stock with a processor that by itself is about the equivalent of the '286 (the 68000). It is *very* close to the limits of not being able to get the jobs the AVT does completed in time. Without a 286 running at 14 Mhz or faster, a PC couldn't do the work, either. If the PC in question doesn't have such a CPU, then one would have to be purchased. Accellerator cards are not all that inexpensive. This isn't as simple as it seems, though. The Amiga *also* comes with three co-processors, dedicated to doing jobs like...

* Display Manipulation;
* Disk Operations;
* Sound Generation;

Memory Size - the Amiga can address, all at one shot, up to 9 million bytes of contiguous memory. The PC's can only do 64 thousand bytes without resorting to "segments". A low res image is almost 50 Kb. A high res image is closer to 200 Kb, and a very high res AVT image is over 400k! That means that the InTel CPU's have to go thru much more work just to get at the image data - never mind processing it.

Parallel Port - the Amiga has a fully functional bi-directional parallel port - the PCs may or they may not. If not, you'd have to buy one, adding perhaps another hundred or so to your cost.

Multitasking - The Amiga comes with extremely efficient multitasking. The PC can have it, but only with OS/2 - and then only at the cost of a *lot* of extra memory - nearly a megabyte *just for the operating system!* The AVT depends on multitasking in many ways.

To sum it up, you (or we) would need to add the following capabilities to a PC to bring it up to a level where it could operate even *close* to what the Amiga can do:

Display - 256 colors, at least
14 Mhz 80286, at least
Multitasking
Dual Channel Audio board, with onboard CPU
Extra memory, 2 mb at least in addition to the 640k
True Parallel port
The AVT system itself

All of these extra cost items for a PC would certainly exceed the cost of buying a one megabyte Amiga 500, a monitor, and the AVT - that's only about $1,400.00 or so.

It's not that we wouldn't *like* to make the AVT system for the millions of PCs and clones out there - but the simple facts are that they don't have what it takes, and bringing them up to that kind of capability is extremely costly. It's much simpler just to buy an Amiga and put it next to the PC. Then you'll have a multitasking, high resolution graphics oriented, speech synthesizing, 4 channel stereo sound generating, windowing computer that does SSTV, FAX, and packet better than anything else on the market... and you'll have the PC, too - if you still want to use it after seeing what an Amiga can do!

Appendix L - Notes On The Development Of The AVT System

The existence of the AVT system is a testimonial to the "pulling yourself up by your bootstraps" philosophy. It is virtually an all-Amiga creation in many ways.

The AVT System uses the 68705 microprocessor in the hardware interface. The assembler for the program that is burned into the 68705 was written for the Amiga by Black Belt Systems specifically for that job. The hardware was designed by Black Belt Systems on the Electron Schematic Capture CAD software which was written for the Amiga by Black Belt Systems. The software was downloaded to the prom burner using an S-record transfer utility written for the Amiga by Black Belt Systems. The CPU programmer, while not a Black Belt Systems product, was designed by Ben Williams, AA7AS when he was working for Logical Devices, Inc. The printed circuit board for the AVT hardware was designed and prototyped by Black Belt Systems using BoardMaster, a PCB CAD program which was written for the Amiga by Black Belt Systems. And of course, the AVT software and the firmware in the 68705 were themselves written by Black Belt Systems.

In short, Black Belt Systems created virtually every tool it used to bring this design to reality as well as the design itself, with the notable exception of the following four excellent development programs:

Lattice's C compiler, version 5.0
MetaComCo's Assembler, version 10.178
Innovatronic's "Power Windows" version 2.5
Rick Stile's "UEDIT"

Once the intial design of the system was complete, testing of a limited number of units began. During this period, AEA noticed the flurry of activity on the SSTV frequencies. They examined what was going on, and decided that this system was an important step up in SSTV/FAX systems. Eventually, they purchased the rights to market and distribute the system from Black Belt Systems. With the knowlege gained from the initial testing, the AVT system was redesigned to work better, faster, and be more flexible in it's hardware capabilities.

The result is the Version Three AVT System - The most flexible SSTV/FAX system made to date, and a real showcase for the Amiga's many extraordinary capabilities.

Appendix M - About The File Requester

The file requester window:

The file requester window provides a point and click environment that allows you to manipulate various file types anywhere within the Amiga's filing system with ease.

In addition to automatically operating when you make the appropriate menu selection, the file requester is available for general use from the scripts and macros that you can run if you have ARexx installed in your machine.

This means (among other things) that you can have completely generalized file specification using this state of the art file requester.

Following is a description of the file requester window.

The file requester has three main areas. These are a **list** area, a **text** area, and a **control button** area.

The **list** area contains three independent lists. Each list is constructed in the same manner.

There are eight available "slots" in a list. To the left of each "slot" is a "select button" that may be pressed to select that item in the list.

To the right of each list is a standard proportional gadget, and two **arrow buttons.** If there are more than eight entries in a particular list, then the knob in the prop gadget will reduce in size, and you will be able to scan the items in the list by moving the knob with your mouse.

Additionally, you may choose to point at and press the arrow buttons, which will move the items through the list one at a time.

Note that the "slot", or the text area, is a string gadget; you can, if a name in the list is too long to see, point and click within the text slot and use the cursor keys to scroll the item through the slot horizontally in order to read it in it's entirety.

The leftmost list is a list of files that are in the currently selected directory. Items in the file list will only show if:

A) They have the same extension as is specified in the "Extension:" field of the text area, or

B) If the "Extension:" field of the text area is blank.

The AVT Master provides default extensions for The AVT Master software automatically places the proper extension in the Extension field as appropriate, so that only files of the correct type will show in the Files list.

The center list is a list of **volumes** that are currently mounted in your Amiga's file system. This list may also display any *Assigned* volume names active in your system if the "Assigns" control button is selected.

The rightmost list is a list of **directories** that are in the currently selected directory.

The **text** area of the file requester contains five individual text gadgets. Four are for your use, and one, the "Messages:" one, is for the file requester's use in telling you what is going on with operations you are asking it to do. Those provided for your use are used as follows:

Path:
This item is used to control where the file requester "looks" for it's lists of directories and files. You can type into this item at any time by pointing into it with the mouse and clicking once. You can also set this item to a particular position in the file system by selecting items from the list of **directories**, items from the list of **volumes**, or by pressing the **parent** control button. If you elect to type into it, make sure you press one of the **enter** or **return** keys on the keyboard when you are done editing the information inside the **path** item. This forces the file requester to try and find that directory in the Amiga's filesystem.

File:
This item is used to determine the actual file that the operation the file requester was invoked to perform will use as it's target. You can fill in this entry by hand, or by pressing a **select** button in the **file** list. If you enter a file name in here and it does not exist in the current directory, then a message to that effect will appear in the "Messages:" item. If you have an extension specified in the "Extension:" field, then the file name will *not* have it's extension in the filename area. If you do not have an extension specified, the file *will* have it's extension.

Extension:
This is an optional field that, if filled in, will force the requester to only show files in the **file** list that have a matching trailing extension. If it is empty, then all files in the directory will show in the **file** list. The AVT Master normally fills in this field for you, and so you should not have to modify it.

Comment:
This item is used to show you comments that are associated with files you select. If you are going to be creating a new file, or writing out an old one, you can enter text into this item and it will be written with the file as a standard AmigaDOS comment.

The **control button** area provides seven different control operations. Each is defined in the following section;

<---->:
The first button will appear with different legends inside it, such as load, save, and so on. Pressing this button will close the file requester and inform the main program that you wish to proceed with the chosen operation, and that you wish to utilize the file you selected. When the file requester is called from ARexx, you supply the name for this button. See the ARexx section for details.

Cancel:
Pressing this button will close the file requester and inform the main program that you wish to CANCEL the chosen operation. The file you chose, if any, is ignored completely.

Parent:
Pressing this button causes the file requester to move "back" one level in the file structure, unless it is already at the root of the volume it is working on.

Delete:
This function allows you to delete individual files from within the AVT Master softwre.

Sort:
This is a "one-shot" sort button. All three lists are sorted alphabetically into ascending order. Using this button can help you to easily find a file or directory when the lists are large and hard to manage.

Sortlock:
This is like the Shift-Lock key on your keyboard; if you prefer to have the sort function always operate when you are using the file requester, then press this button. Each time the file requester is brought up, the lists will be sorted for you. Pressing the button again will turn off the function.

Assigns:
This button is also a lock type function. It changes the center list from showing only volumes to showing volumes and "assigns". This can be handy if your system is heavily salted with assigned volume names. Pressing the button again will turn off the function.

Low Memory Conditions:

If you are extremely low on chip memory, the main file requester may be unable to open. If this happens, a small text requester will appear instead. You must type the full path, filename and extension into this requester and the system will attempt to handle the file this way.

This will *only* occur if you are very low on memory, chip memory specifically. If it does occur, we recommend that you obtain additional memory for your Amiga as soon as possible.

Glossary

68000

The 68000 is the part number of the microprocessor used in stock Amiga 500's, 1000's, and 2000's. This microprocessor was designed by Motorola Corporation, and offers 16 bit data handling to and from memory and peripherals, while all internal operations are 32 bits wide. It is a very high performance microprocessor, much more so than the ones used in the clone machines such as the InTel 8088 and the 8086.

68020

The 68020 is the part number of the microprocessor used in stock Amiga 2500's, and in some modified (accelerated) Amiga 500's, and 1000's. The 68020 offers 32 bit data paths both inside and out, and generally operates between two and four times as fast as a 68000, assuming the same frequency system clock. Typical speeds for 68020 processors in Amiga's are in the 14 MHz range. These 68020's can be thought of as comparable to an Intel 80386 processor running at about 20 MHz, and far superior to an 80286 at any speed.

68030

The 68030 is a microprocessor that, at the time of this writing, is only available for the Amiga from third party (aftermarket) entities. This microprocessor is the most sophisticated mass-production part currently available from any manufacturer. In addition to being more efficient than a 68020, that is, at the same clock, things get done faster, it can run even faster as well. One Amiga add-on board has a 33 MHz 68030 in it, resulting in a blindingly fast system!

a500

The Amiga 500 is the "entry-level" computer in the Amiga series. It is the lowest cost unit, and offers complete software compatibility with all the other Amiga systems. The 500 is not as expandable as the 2000 or 2500 units, however, and cannot use the same hardware add-ons (like serial ports, hard drive controllers, and so on) that these units can. It does have a line of peripherals that work with it, and is a very good system in its own right. One major disadvantage of the 500 is that the keyboard is part of the computer's case, and so the whole system must be moved when you readjust the keyboard to suit your preferences.

a1000

The Amiga 1000 is the original Amiga system. It is no longer manufactured, however they can occasionally be found on the used market. The 1000 is a very attractively styled unit, often considered the most attractive of all the available systems with its low profile cabinet and keyboard "garage". The main drawback to owning a 1000 is that the hardware expansion of this system is becoming difficult, as few manufacturers are creating peripherals for it anymore.

a2000

The Amiga 2000 is the *ideal* Amiga system to purchase at the time of this writing. It offers the most expansion capability, with many internal slots for plug-in cards of various types. It is more expensive than the a500, but is well worth it in this writers opinion. In addition to the normal expansion slots, the 2000 offers IBM/clone compatible slots so that you can have a complete 8088 or 80286 IBM computer *inside* the case of the Amiga! Two computers in the space of one, in fact. The 2000 offers a detachable keyboard as well.

a2500

The Amiga 2500 is the current "top of the line" performer from CBM. It uses a 68020 processor running at 14+ MHz, and comes with either two or four megabytes of 32 bit wide RAM, as well as one megabyte of 16 bit wide RAM. Basically an Amiga 2000 with a different processor and wider memory, this machine is the most powerful complete microprocessor system you can buy in the under $10,000 price range.

AM FAX

These are facsimile images that encode the brightness of each pixel by the loudness of the FAX signal. You will find FAX signals of this type coming from satellites in the NOAH and GEOS series, as well as the Russian COSMOS series. The AVT system is not able to receive these types of FAX signals unless you add the Overview Systems board. See appendix G for details on this product.

Analog

This is a type of electronics that process information and data using linear techniques. Generally more prone to noise and error, an analog signal can carry much more information than a digital signal, and so for some types of error-resistant data transfers (like images!) analog techniques are superior.

ARexx

ARexx is probably one of the single most important software packages ever to be released for the Amiga computer. It offers stand alone program capability like "Basic", scripting like the CLI startup sequence scripts, macros like those found in editors and spreadsheets, and facilitates communications between multiple programs within the Amiga system. Available from your dealer and soon to be included with the Amiga's operating system (AmigaDOS version 1.4, information received from Commodore as of August 1989), ARexx allows you do tricks you never thought were possible. We strongly recommend that you obtain a copy of ARexx to use with the AVT system and your other Amiga software.

Assembly

Assembly (language) is the language used to write programs in the actual language used by the microprocessor - so-called "machine language". It is difficult to write, more so than any other language, because humans and microprocessors tend to do things quite differently. It is used because when programs are written in it, they are generally much, much faster than if they were written in a "high level" language such as C, Basic, or Modula. About 25% of the AVT system is written in Assembly language, mostly the receive and transmit portions.

AVT

AVT stands for "Amiga Video Transceiver", and is the name of both the system (the AVT System) and a shorthand way of referring to the new modes that the system uses to transfer SSTV images.

b2000

The Amiga b2000 is the same as the Amiga a2000 except for some internal differences. One was made in Europe and the other in WestChester PA a little later (and a little better, by all reports). See the Glossary entry on the a2000 for more background.

Black Belt System

This is the company that designed the AVT system and licensed it to AEA for marketing and manufacture. Located in Glasgow, Montana, in the USA, Black Belt Systems is a company that was formed specifically to cater to the needs of the Amiga technical community and the Amiga ham radio operator.

Black Belt Systems
RR1, Box 4272
398 Johnson Road
Glasgow, MT
59230
406-367-5509

C

C is a "high-level" language that is compiled into machine language for use by the computer. C is generally one of the best performing of the high level languages. All of the portions of the AVT Master software that are not written in assembly language were written in C.

Chromanance

This is the portion of a composite signal that carries the color information.

Composite

This is a term that means that color and black and white information is combined in such a way that (1) a black and white receiver can still recover the black and white image, and (2) a color receiver "sees" a color image.

Digital

As opposed to analog, this is a means whereby data (like images or radioteletype) is transferred such that it can be checked for errors at the receiving end. information is encoded in binary (base two) as either ones or zeros, and sent that way. Because of this, a digital transmission generally carries much less information than an analog transmission, though with more reliability and repeatability.

DX

This term literally translates to "distance", and is used by ham radio operators and shortwave listeners (SWLs) the world over. To say that a radio station is a DX station is to say that it is far away and difficult to receive, or possibly it would be *unlikely* that you would receive it because the station might be rarely on the air or operating with extremely low power.

FAX

FAX is short for facsimile. This is a method of image transmission that sends black and white images in fairly high resolution over periods of ten minutes or so. Recently, "office FAX" has appeared, and people often confuse this with the original FAX modes. Office FAX is a digital mode, similar to the data that a computer's modem sends over the phone line. The original FAX modes are analog modes, sending far more data in the same amount of time.

FCC

The FCC is the Federal Communications Commission. This is the federal agency charged with seeing to it that the radio spectrum is not misused or abused. They provide the regulations (and the penalties!) that govern the design and manufacture of devices like the AVT system that generate and/or use RF energy. The FCC's participation in the approval of a product for release to the consumer is a factor that vastly increases the cost of creating such a product. While they certainly perform a useful service in helping to keep the radio spectrum clean, the manner in which they go about it makes it very difficult to get a product to market at a reasonable cost and within a reasonable time. Write your congressperson!

FM FAX

These are facsimile images that encode the brightness of each pixel by the frequency of the FAX signal. You will find FAX signals of this type coming from radio station NAM in Norfolk, Virginia, as well as other outlets transmitting in the shortwave spectrum (3-30 MHz). These images are directly compatible with the AVT System.

Font

A Font, in the Amiga context, is a group of character shapes that have a particular "look" or "feel" to them. For instance, Ye Olde Englishe Lytering as typically found on "Pub" signs would be considered a font. So would letters formed in a manner similar to handwritten script, or letters that look like they came off of a typewriter. Fonts come in sizes, referred to as point sizes. The AVT can use Amiga fonts from five to 50 points, inclusive.

Frame

A Frame is one screen or complete image of an SSTV picture. A frame may or may not take up the entire screen, but will always represent the entire image. A color frame can also be thought of as having three individual frames, the red, green, and blue frames.

HAM Images

The term HAM here stands for Hold-And-Modify, and refers to one of the Amiga's graphics display modes. A HAM image can contain up to 4096 color and brightness levels simultaneously, unlike the other modes which at best offer 64 simultaneous colors/brightnesses from a palette of 4096. This is also in sharp contrast to other computer systems such as the Mac and the clones, none of which come standard with such a rich color capability. The Amiga does - of course!

Ham Radio

The term Ham here seems to be a contraction of the term "Amateur", as in Hamatuer. People who operate Ham radio refers to themselves as Hams, more often than not, and quite often you'll find a drawing of a pig somewhere in their "shack", which is where they keep their radios and generally pursue their hobby. To become a Ham radio operator requires that you learn some basic electronics, how to receive (copy down on paper) morse code, and take tests on both of these items. Once this is done, you may communicate with other Hams the world over, which is fun, educational, and offers many opportunities to provide public services such as disaster relief and emergency communications, message passing, and more.

Histogram

The histogram is a graph that represents the distribution of pixels of various brightnesses throughout an image. An image that is "well composed" often has a bell-shaped curve, a Gaussian distribution. Histograms have been in use in the image processing field for a number of years, and have proved their worth many times over.

Luminance

This is the portion of a composite signal that represents the black and white, that is, the Greyscale information in an image. It can be used without the color information if a black and white image is acceptable.

Overview

The Overview board is a hardware add-on for the AVT System that provides AM FAX reception capability. It does this by converting AM FAX to FM FAX, as well as providing other unique features. For details, refer to **Appendix G.**

Pixel

A pixel is one location on the computer's display that is the smallest area that can be set to a particular grey level or color. The AVT System uses displays that have pixel dimensions of 320 across by 240 down, 320 across by 400 down, and 640 across by 400 down.

QRM

This is an abbreviation that means man-made noise or interference, whether intentional or not. It comes from Ham radio operator abbreviations. Some examples might be a station that is talking on the same frequency you are trying to transfer images, interference generated from incorrectly adjusted linear amplifiers, RFI from computer systems, and so on.

QRN

This is an abbreviation that means natural noise (as opposed to man-made) or interference. Examples are crackles and pops from electrical activity within thunderstorms (such as lightning), static electricity noise from walking on a rug, hiss, and any number of strange noises that waltz out of the atmosphere from time to time.

Resolution

Resolution refers to the detail of any component of an image. For instance, you can speak of color resolution; The Amiga has a maximum color resolution of 4,096 simultaneous colors when operating in HAM mode. You can also speak of spatial resolution; The Amiga has a spatial resolution of 320 by 400 when operating in the HAM mode. When dealing with SSTV and FAX, you can also speak of FM resolution; The AVT system has a 64 level FM resolution across the grey levels, and a 256 level resolution across its entire FM range.

RFI

This stands for Radio Frequency Interference. It means any signal that is generated by a device that should not be generating it, within the range of radio frequencies.

Robot

This is both the name of a company (Robot Research, Inc.) and shorthand for any of their products. You might hear it in this context: "I'm using a Robot". It has nothing to do with the classical meaning of the word such as a self-motivating mechanism.

Speed

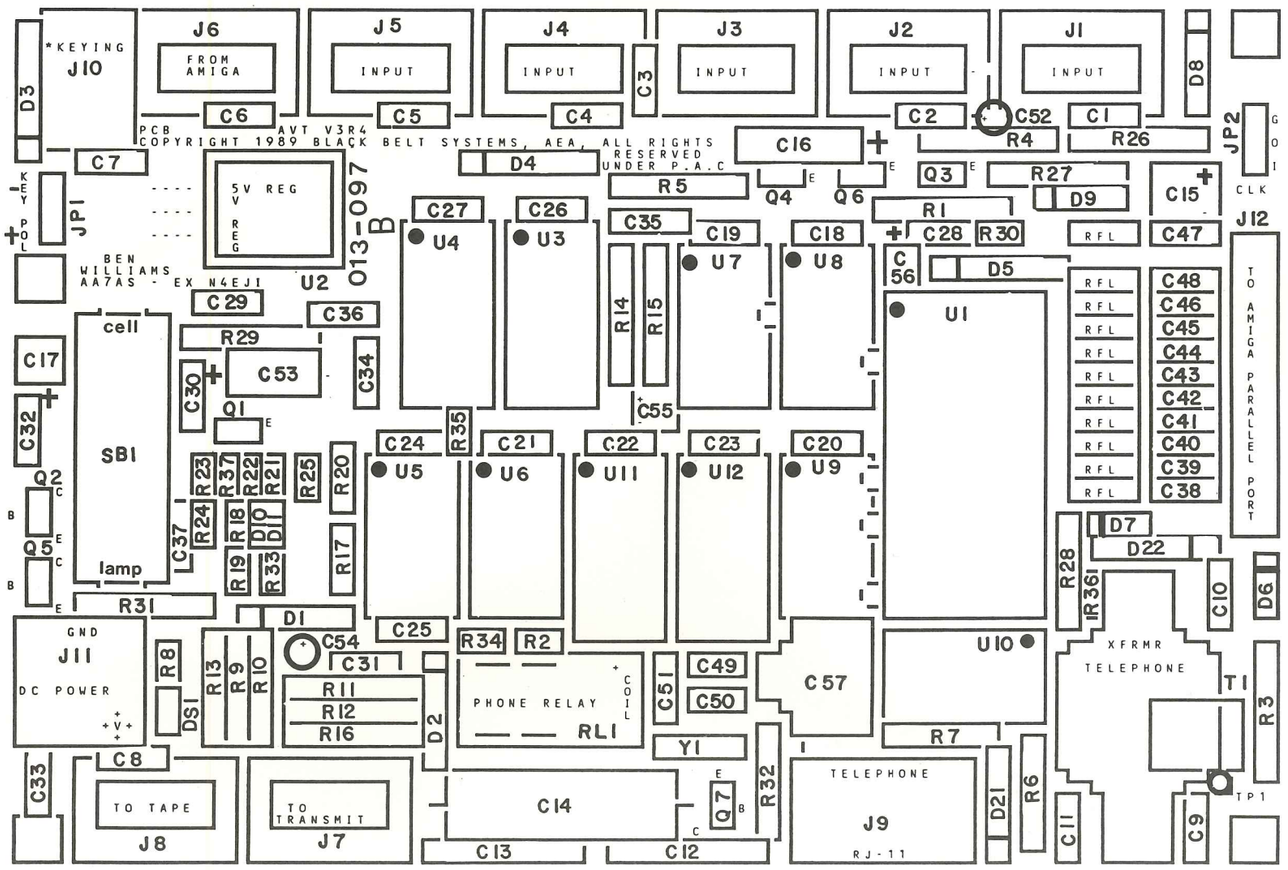
In the SSTV and FAX context, the word speed indicates how quickly a complete image is transferred. We speak of eight-second speeds, and 36-second speeds, and so on. What we mean by this is an eight-second total frame transmission time, or a 36-second total frame time.

SSTV

This stands for Slow Scan TeleVision. The term was first used by the small group of amateurs led by Mr. Cop McDonald some time ago when they pioneered the concept of video images by audio frequency over HF radio. Although the initial efforts were extremely crude, and suffered from poor design and various technical shortcomings more related to the components available than the designers, SSTV caught on because it is a terrific way to share the Amateur hobby.

Index

Component Location



AVT MANUAL 040-960
Rev. A. 10/89

AVT Amiga Video Terminal Operating Manual

ADVANCED ELECTRONIC APPLICATIONS, INC.

AVT SYSTEM

Amiga Video Terminal

Operating Manual